Advances in Intelligent Systems and Computing 733

Sheng-Lung Peng
Shiuh-Jeng Wang
Valentina Emilia Balas
Ming Zhao *Editors*

Security with Intelligent Computing and Big-data Services

Springer

Advances in Intelligent Systems and Computing

Volume 733

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Advances in Intelligent Systems and Computing" contains publications on theory, applications, and design methods of Intelligent Systems and Intelligent Computing. Virtually all disciplines such as engineering, natural sciences, computer and information science, ICT, economics, business, e-commerce, environment, healthcare, life science are covered. The list of topics spans all the areas of modern intelligent systems and computing.

The publications within "Advances in Intelligent Systems and Computing" are primarily textbooks and proceedings of important conferences, symposia and congresses. They cover significant recent developments in the field, both of a foundational and applicable character. An important characteristic feature of the series is the short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

More information about this series at http://www.springer.com/series/11156

Sheng-Lung Peng · Shiuh-Jeng Wang
Valentina Emilia Balas · Ming Zhao
Editors

Security with Intelligent Computing and Big-data Services

Springer

Editors
Sheng-Lung Peng
Department of Computer Science
and Information Engineering
National Dong Hwa University
Hualien
Taiwan

Valentina Emilia Balas
Department of Automation and Applied
Informatics, Faculty of Engineering
Aurel Vlaicu University of Arad
Arad
Romania

Shiuh-Jeng Wang
Department of Information Management
Central Police University
Taoyuan
Taiwan

Ming Zhao
School of Computer Technology
Jingzhou
China

ISSN 2194-5357 ISSN 2194-5365 (electronic)
Advances in Intelligent Systems and Computing
ISBN 978-3-319-76450-4 ISBN 978-3-319-76451-1 (eBook)
https://doi.org/10.1007/978-3-319-76451-1

Library of Congress Control Number: 2018935887

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG
part of Springer Nature
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The purpose of 2017 International Conference on Security with Intelligent Computing and Big-data Services (SICBS'17 for short) with joined workshops, Workshop on Information and Communication Security Science and Engineering and Workshop on Security in Forensics, Medical, and Computing Services and Applications, is to provide a platform for researchers, engineers, academicians, as well as industrial professionals from all over the world to present their research results and development activities in security-related areas. It also aims at strengthening the international academic cooperation and communications and exchanging research ideas.

It is the first SICBS. We wish that it could be then kept continued to the second SICBS as this 2017 successful academic activity. This SICBS 2017 brought together researchers from all regions around the world working on a variety of fields and provided a stimulating forum for them to exchange ideas and report on their progress in researches. In the conference including the two workshops, we collect 34 papers, covering the topics as follows: Algorithms and Security Analysis, Cryptanalysis and Detection Systems, IoT and E-commerce Applications, Privacy and Cloud Computing, Information Hiding and Secret Sharing, Network Security and Applications, Digital Forensics and Mobile Systems, Public Key Systems and Data Processing, and Blockchain Applications in Technology.

Organization of conferences is a hard work. It would not have been possible without the exceptional commitment of many expert volunteers. We would like to thank all those who contributed to the advisory committee, the technique program committee, and the organizing committee for their efforts in the course of conference preparations. We also give our most thanks to all the authors of the submitted papers to make this conference successful in the good paper quality for presentations. We are grateful to Springer for publishing the proceedings.

Finally, but not the least, we hope that the participants will not only enjoy the technical program during this prestigious conference but also discover many beautiful attractions in Taroko National Park, in particular Qingshui Cliff, a top-ten scenic area in Taiwan to make their stay unforgettable. Wishing you a fruitful and enjoyable SICBS 2017!

Sheng-Lung Peng
Shiuh-Jeng Wang
Valentina Emilia Balas
Ming Zhao

Organization

International Conference on Security with Intelligent Computing and Big-data Services (SICBS 2017)

Honorary Chairs

Chin-Chen Chang	Feng Chia University, Taiwan
Han-Chieh Chao	National Dong Hwa University, Taiwan

General Chairs

Sheng-Lung Peng	National Dong Hwa University, Taiwan
Shiuh-Jeng Wang	Central Police University, Taiwan

International Advisory Committee

Lakhmi C. Jain	University of Canberra, Australia
Marc Joye	NXP Semiconductors, San Jose, CA, USA
Hiroaki Kikuchi	Meiji University, Japan
Kwangjo Kim	KAIST, Korea
Nakao Koji	KDDI/NICT, Japan
Sakurai Kouichi	Kyusyu University, Japan
Prabhat Kumar	Nation Institute of Technology Patna, India
Javier Lopez	University of Malaga, Spain
Witold Pedrycz	University of Alberta, Canada
Laurence T. Yang	St. Francis Xavier University, Canada
Sang-Soo Yeo	Mokwon University, Korea
Heung Youl Youm	Soonchunhyang University, Korea

Program Chairs

Nilanjan Dey	Techno India College of Technology, India
Ching-Nung Yang	National Dong Hwa University, Taiwan
Taeshik Shon	Ajou University, Korea

Publicity Chairs

Cheng-Ta Huang	Oriental Institute of Technology, Taiwan
Da-Yu Kao	Central Police University, Taiwan
Seungmin Rho	Sungkyul University, Korea
Aneesh Sharma	University of California, Berkeley, USA
Gulshan Shrivastava	National Institute of Technology Patna, India
Ting-Ting Yang	National University of Tainan, Taiwan

Publication Chairs

Valentina Emilia Balas	Aurel Vlaicu University of Arad, Romania
Ming Zhao	Yangtze University, China

Organizing Chairs

Changpo Chiang	Central Police University, Taiwan
Jongsung Kim	Kookmin University, Korea
Damien Sauveron	University of Limoges, France
Shiow-Yang Wu	National Dong Hwa University, Taiwan

Organizing Committee

I-Cheng Chang	National Dong Hwa University, Taiwan
Tao-Ku Chang	National Dong Hwa University, Taiwan
Min-Xiou Chen	National Dong Hwa University, Taiwan
Han-Ying Kao	National Dong Hwa University, Taiwan
Pao-Lien Lai	National Dong Hwa University, Taiwan
Guan-Ling Lee	National Dong Hwa University, Taiwan
Shou-Chih Lo	National Dong Hwa University, Taiwan

Technical Program Committee

Jinsuk Baek	Winston-Salem State University, USA
Yung-Kuan Chan	National Chung Hsing University, Taiwan
Chi-Chao Chang	Chang Jung Christian University, Taiwan
Chun-Young Chang	National Police Agency, Taiwan

Wei Ping Chang	Central Police University, Taiwan
Ya-Fen Chang	National Taichung University of Science and Technology, Taiwan
Yue-Shan Chang	National Taipei University, Taiwan
Chia-Mei Chen	National Sun Yat-sen University, Taiwan
Chien-Ming Chen	Harbin Institute of Technology, China
Chien-Yuan Chen	National University of Kaohsiung, Taiwan
Da-Ren Chen	National Taichung University of Science and Technology, Taiwan
Hsiang-Lin Chen	National Police Agency, Taiwan
Hsing-Chung Chen	Asia University, Taiwan
I-Te Chen	Kaohsiung Medical University, Taiwan
Sheng-Wei Chen	Academia Sinica, Taiwan
Te-Yu Chen	National Tainan Junior College of Nursing, Taiwan
Tung-Shou Chen	National Taichung University of Science and Technology, Taiwan
Tzer-Shyong Chen	Tung Hai University, Taiwan
Tzung-Her Chen	National Chiayi University, Taiwan
Wei-Kuei Chen	Chien Hsin University of Science and Technology, Taiwan
Yen-Wen Chen	National Central University, Taiwan
Yi-Hui Chen	Asia University, Taiwan
Yi-Ming Chen	National Central University, Taiwan
Yong-Sheng Chen	National Taipei University of Education, Taiwan
Yu-Chi Chen	Yuan Ze University, Taiwan
Yu-Yi Chen	National Chung Hsing University, Taiwan
Bo-Chao Cheng	National Chung Cheng University, Taiwan
Chen-Mou Cheng	National Taiwan University, Taiwan
Shin-Ming Cheng	National Taiwan University of Science and Technology, Taiwan
Hung-Yu Chien	National Chi Nan University, Taiwan
Naveen Chilamkurti	La Trobe University, Australia
Wen-Long Chin	National Cheng Kung University, Taiwan
Chao-Lung Chou	National Defense University, Taiwan
Chih-Ho Chou	National Center for Cyber Security Technology, Taiwan
Li-Der Chou	National Central University, Taiwan
Shih-Su Chou	Coast Guard Administration, Taiwan
Yao-Hsin Chou	National Chi Nan University, Taiwan
Yung-Chen Chou	Asia University, Taiwan
Hai-Cheng Chu	National Taichung University of Education, Taiwan
Chung-Mei Fan	Taiwan Hospital Association, Taiwan
Wei-Hua He	Soochow University, Taiwan

Sheng-Chih Ho	National Defense University, Taiwan
Shuyuan Mary Ho	Florida State University, USA
Gwoboa Horng	National Chung Hsing University, Taiwan
Hsu-Chun Hsiao	National Taiwan University, Taiwan
Pao-Ann Hsiung	National Chung Cheng University, Taiwan
Fu-Hau Hsu	National Central University, Taiwan
I-Ching Hsu	National Formosa University, Taiwan
Yu-Chen Hu	Providence University, Taiwan
Cheng-Ta Huang	Oriental Institute of Technology, Taiwan
Chun-Ying Huang	National Chiao Tung University, Taiwan
Shih-Kun Huang	National Chiao Tung University, Taiwan
Min-Shiang Hwang	Asia University, Taiwan
Ren-Junn Hwang	Tamkang University, Taiwan
Shin-Jia Hwang	Tamkang University, Taiwan
Ji-Han Jiang	National Formosa University, Taiwan
Wen-Shenq Juang	National Kaohsiung First University Science and Technology, Taiwan
Cheonshik Kim	Sejong University, Korea
Tung-Ming Koo	National Yunlin University of Science and Technology, Taiwan
Wei-Chi Ku	National Taichung University of Education, Taiwan
Fu-Chung Kuo	National Police Agency, Taiwan
Rajesh Laskary	Jodhpur Engineering College & Research Centre, Singapore
Cheng-Chi Lee	Fu Jen Catholic University, Taiwan
Chin-I Lee	Ling Tung University, Taiwan
Ching-Feng Lee	Chaoyang University of Technology, Taiwan
Jung-San Lee	Feng Chia University, Taiwan
Narn-Yih Lee	Southern Taiwan University of Science and Technology, Taiwan
Shao-Lun Lee	Oriental Institute of Technology, Taiwan
Wei-Bin Lee	Feng Chia University, Taiwan
Yih-Jiun Lee	Private Chinese Culture University, Taiwan
Chun-Ta Li	Tainan University of Technology, Taiwan
Jung-Shian Li	National Cheng Kung University, Taiwan
Ming-Fu Li	Chang Gung University, Taiwan
You-Lu Liao	Central Police University, Taiwan
Ally Chia-Chen Lin	Providence University, Taiwan
Chu-Hsing Lin	Tung Hai University, Taiwan
Feng-Tse Lin	Chinese Culture University, Taiwan
Han-Yu Lin	National Taiwan Ocean University, Taiwan
Hsi-Chung Lin	Aletheia University, Taiwan
Iuon-Chang Lin	National Chung Hsing University, Taiwan
Jin-Cherng Lin	Tatung University, Taiwan

Kai-Biao Lin	Lecturer, Xiamen University of Technology, China
Po-Ching Lin	National Chung Cheng University, Taiwan
Pei-Yu Lin	Yuan Ze University, Taiwan
Tsung-Hung Lin	National Chin-Yi University of Technology, Taiwan
Yu-Li Lin	Ministry of Justice Investigation Bureau, Taiwan
Jigang Liu	Metropolitan State University, Minnesota, USA
Jonathan C. L. Liu	University of Florida, USA
Der-Chyuan Lou	Chang Gung University, Taiwan
Cheng-Yu Lu	104 Corporation, Taiwan
Chung-Fu Lu	Chihlee Institute of Technology, Taiwan
Tzu-Chuen Lu	Chaoyang University of Technology, Taiwan
Jia-Ning Luo	Ming Chuan University, Taiwan
Masahiro Mambo	Kanazawa University, Korea
Atsuko Miyaji	Osaka University, Japan
Uramoto Naohiko	Former IBM-Tokyo-Lab, Japan
Chung-Ming Ou	Kainan University, Taiwan
Hsia-Hung Ou	National Taiwan Sport University, Taiwan
Hua-Wang Qin	Nanjing University of Science & Technology, China
Aneesh Sharma	University of California, Berkeley, USA
Jau-Ji Shen	National Chung Hsing University, Taiwan
Chin-Shiuh Shieh	National Kaohsiung University of Applied Sciences, Taiwan
Dongkyoo Shin	Sejong University, Korea
Ming-Jhih Siao	Tri-Service General Hospital, Taiwan
Ching-Wei Su	National Police Agency, Taiwan
Wei-Liang Tai	Chinese Culture University, Taiwan
Yi-Lang Tsai	National Applied Research Laboratories, Taiwan
Woei-Jiunn Tsaur	National Taipei University, Taiwan
Tzu-Liang Tseng	University of Texas at El Paso, USA
Yuh-Min Tseng	National Changhua University of Education, Taiwan
Hao-Kuan Tso	Chien Hsin University of Science and Technology, Taiwan
Ray-Lin Tso	National Cheng-Chi University, Taiwan
Chen-Kun Tsung	National Chin-Yi University of Technology, Taiwan
Wen-Guey Tzeng	National Chiao Tung University, Taiwan
Danilo V. Vargas	Kyusyu University, Japan
M. Vijayalakshmi	Thiagarajar College of Engineering, Madurai, India
Chih-Hung Wang	National Chiayi University, Taiwan

Ching-Te Wang	National Chin-Yi University of Technology, Taiwan
Ping Wang	Kun Shan University, Taiwan
Wei-Jen Wang	National Central University, Taiwan
Kuo-Jui Wei	AAA Security Technology Co. Ltd., Taiwan
Tai-Kuo Woo	National Defense University, Taiwan
Hsien-Chu Wu	National Taichung University of Science and Technology, Taiwan
Hsin-Lung Wu	National Taipei University, Taiwan
Mu-En Wu	National Taipei University of Technology, Taiwan
Tzong-Sun Wu	National Taiwan Ocean University, Taiwan
Wen-Chuan Wu	Aletheia University, Taiwan
Yi-Chao Wu	Chihlee Institute of Technology, Taiwan
Yunbing Wu	Fuzhou University, China
Liudong Xing	University of Massachusetts, Dartmouth, USA
Cheng-Hsing Yang	National Pingtung University, Minsheng Campus, Taiwan
Kai-Sheng Yang	National Police Agency, Taiwan
Ming-Hour Yang	Chung Yuan Christian University, Taiwan
Ting-Ting Yang	National University of Tainan, Taiwan
Hao-Zhen Ye	Fuzhou University, China
Jieh-Shan Yeh	Providence University, Taiwan
Kuo-Hui Yeh	National Dong Hwa University, Taiwan
Chih-Hao Yen	National Police Agency, Taiwan
Xun Yi	RMIT University, Australia
Hsin-Ming Yu	Chang Gung Memorial Hospital, Taiwan
Mingwu Zhang	Hubei University of Technology, China
Shun-Zhi Zhu	Xiamen University of Technology, China

Contents

Algorithms and Security Analysis

Cryptanalysis and Detection Systems

IoT and E-commerce Applications

Privacy and Cloud Computing

Information Hiding and Secret Sharing

Network Security and Applications

Digital Forensics and Mobile Systems

Public Key Systems and Data Processing

Blockchain Applications in Technology

Algorithms and Security Analysis

A Social Tagging Recommendation Model Based on Improved Artificial Fish Swarm Algorithm and Tensor Decomposition

Hao Zhang[1,2](✉), Qiong Hong[3], Xiaomeng Shi[1], and Jie He[1]

[1] School of Transportation, Southeast University, Nanjing 210096, China
andyhao@seu.edu.cn
[2] Faculty of Transportation Engineering, Huaiyin Institute of Technology,
Huai'an 223003, China
[3] Business School, Huai'an Institute of Information Technology, Huai'an 223003, China

Abstract. Folksonomy Tag Application (FTA) has emerged as an important approach of Internet content organization. However, with the massive increase in the scale of data, the information overloading problem has been more severe. On the other hand, traditional personalized recommendation algorithms based on the interaction between "user-item" are not easy to extend to the three dimensional interface of "user-item-tag". This paper proposes a clustering analysis method for the initial dataset of the Tag Recommendation System (TRS) based on the improvement of Artificial Fish Swarm Algorithm (AFSA). The method is used for dimension reduction of TRS datasets. To this end, considering the weight of the elements in TRS and the score that can reveal user preference, a novel weighted tensor model is established. And in order to complete the personalized recommendation, the model is solved by the tensor decomposition algorithm with dynamic incremental updating. Finally, a comparative analysis between the proposed FTA algorithm and the two classical tag recommendation algorithms is conducted based on two sets of empirical data. The experimental results show that the FTA algorithm has better performance in terms of the recall rate and precision rate.

Keywords: Artificial fish swarm algorithm · Clustering analysis
Tensor decomposition · Tag recommendation

1 Introduction

In traditional Personalized Recommendation Systems (PRS), context-based recommendations are often realized by utilizing the properties of items (e.g. documents, commodities, services). Alternatively, collaborative filtering recommendations are carried out through exploiting the similarities among users and items. One of the primary characteristics of the latter approach is the usage of the two–dimensional (2D) pairwise interactions between users and items (i.e. user-user, user-item, item-item) to filter and analyze the data. Related data analytical techniques include Vector Space Model (VSM), Term Frequency–Inverse Document Frequency (TF/IDF) and some Natural Language

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 3–13, 2018.
https://doi.org/10.1007/978-3-319-76451-1_1

Processing (NLP) machine learning algorithms etc. [1]. As the advancement of Web 2.0, socialized tag has gradually turned into an important technique to organize the context of Internet. An application of socialized tag technique namely Folksonomy tag recommendation has become one of the top popular research topics in personalized recommendation field.

A typical Socialized Tag Recommendation System (STRS) contains three fundamental elements i.e. user, item and tag. And it extends the traditional two-dimensional relations to three-dimension (3D). Through the modeling and analysis of the higher dimensional data, richer information is provided to the recommendation system. However, at the same time, a new issue occurs resulting from the information enrichment, i.e. traditional algorithms are unable to handle with such data directly. To this end, some researches regarded the "user-item-tag" relations as a Tripartite Graph, and decomposed the graph to three Bipartite Graphs while performing recommendation tag system. Then, conventional recommendation algorithms (e.g. collaborative filtering algorithm) were integrated to recommend the tag or item [2]. Meanwhile, other scholars such as Liao et al. (2012) attempted to explore a method to balance the tradeoff between the maximum retention of 3D relations information and the reduction of complexity in terms of handling with high dimensional sparse data [3]. And based on the Tripartite Graph, they proposed a tensor 3D decomposition algorithm and their algorithms were validated through case studies. The key idea of Tripartite Graph is the conversion of 3D relations in socialized tag system to 2D relations. Although the complexity of the system is reduced, the latent information among elements in tag system is missing.

Moreover, the add-on of tag data increases the scale and sparse issues of the recommendation system. Symeonidis et al. (2009) applied tensor decomposition theory into tag recommendation system [4]. They adopted tensor decomposition method to predict the tag so as to reduce the sparse degree of the data. Likewise, to handle with the data scale and noise issues arising from users' random label, Sun et al. (2012) and Wang et al. (2015) clustered the elements in tag system [5, 6]. Moreover, plenty of effective researches on socialized tag system were conducted through the mining of association rules, the analysis of link structure and the development of probabilistic models [7].

The novel contributions of this paper is described as follows: Firstly, a new type of swarm intelligence algorithm – Artificial Fish Swarm Algorithm (AFSA) is adopted to cluster the elements in tag system [8]. As such, the dimension of initial input data for the socialized tag recommendation model is reduced. Then, the tensor decomposition algorithm with dynamic incremental updating is utilized to model tag recommendation system. This step can achieve this function of maximize the latent relations among the elements in tag system while satisfying the performance of recommendation system. The efforts of this study can contribute on the quality improvement of socialized recommendation systems as well as providing high-quality recommendation services to users.

2 Data Clustering Based on Improved AFSA

2.1 AFSA and Its Improvement

The artificial fish swarm algorithm (AFSA) is a random autonomous optimization model proposed according to the characteristics of fish activities. Its basic idea is that within a pool of water, the region with the largest number of local fish is usually the most abundant nutrients area. Therefore, given this assumption, we can simulate the fish's feeding, cluster and following behavior so as to achieve the goal of global optimization [8]. In this paper, the basic AFSA is improved to cluster the initial data set of tag recommendation system. Further, the optimal cluster results are used as the input data for the incremental tensor decomposition algorithm in the next step. The parameter description and several typical behaviors of the artificial fish swarm algorithm are as listed as follows (we take the maximum value case as an example).

(1) Parameter description: N is the total number of artificial fish and $X = (x_1, x_2, \ldots, x_n)$ represents the individual state of artificial fish, $x_i (i = 1, 2, \ldots, n)$ is the non-optimized variable. $Y = f(X)$ represents the food concentration (i.e., fitness function) of the artificial fish in the current position m (Y is the value of the objective function). *Visual* is the visual perception distance of artificial fish, δ is crowding factor, *Try_number* is tentative times. And the distance between individual fish i and j is $d_{ij} = \left| x_i - x_j \right|$, *Step* represents the step length, *Rand* () generates (0-1) random numbers.

(2) Preying behavior: *Prey* (), X_i is the current state of artificial fish, Y_i is the current fitness, and X_j is a randomly selected state within its visual perception distance. If $Y_i < Y_j$, then go forward in this direction and update the current state. Else, reselect random state X_j and judge the state update condition and repetitive operation after repeated the times of *Try_number*, which still fails to meet the conditions. And then moves one step at random. Its preying rules are defined as the Eq. (1).

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^{\mathrm{t}}}{\left\| X_j - X_i^{\mathrm{t}} \right\|} \cdot S\mathrm{tep} \cdot \mathrm{Rand}(), Y_i < Y_j \tag{1}$$

(3) Swarming behavior: *Swarm*(), X_i is the current state of artificial fish, search the number of partners (n_f) in the current neighborhood (d_{ij} < *Visual*) and the center position X_c. If $Y_c / n_f > \delta Y_i$. It indicates that the partner center has more food and is not crowded. Then moves one step to the center position of the partner, and its moving rule is defined as the Eq. (2), otherwise perform preying behavior.

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^{\mathrm{t}}}{\left\| X_c - X_i^{\mathrm{t}} \right\|} \cdot S\mathrm{tep} \cdot \mathrm{Rand}(), Y_c / n_f > \delta Y_i \tag{2}$$

(4) Following behavior: *Follow*(), X_i is the current state of artificial fish, search partner X_{max} in the nearest neighborhood ($d_{i\,j}$ < *Visual*) with the largest fitness

value Y_j. If Y_j /n_f > δY_i, it indicates that partner X_{max} ‘s current state has a high food concentration and is not crowded around. It advances one step to partner X_{max} in the direction, and its moving rule is defined as the Eq. (3), otherwise perform preying behavior.

$$X_i^{t+1} = X_i^t + \frac{X_{\max} - X_i^{\mathrm{t}}}{\|X_{\max} - X_i^{\mathrm{t}}\|} \cdot Step \cdot \mathrm{Rand}(), Y_j/n_f > \delta Y_i \tag{3}$$

(5) Behavior strategy and bulletin board, artificial fish will choose the best behavior strategy according to their own state and surrounding environment so that they can reach the higher concentration of food and improve the efficiency of optimization. At the same time, the algorithm sets up a bulletin board to record the current state and the current optimal food concentration values for each artificial fish after the corresponding action update.

AFSA has good ability to jump out of local extremum and realize global adaptive search. It shows good adaptability in many fields such as parameter optimization, data clustering and signal processing [8]. However, the AFSA has the disadvantages of large search blindness, slow convergence speed and low optimization accuracy in the later stage of operation. Therefore, by using the idea of literature [9, 10], we proposed an adaptive step size algorithm to reduce the later search blindness and improve the speed of convergence. Through the behavior choice of strategies and the bulletin board, the artificial fish location update strategy is optimized. The detailed improvement methods are as follows:

(1) Improvement of adaptive visual field and step size of artificial fish. The visual field and step size of artificial fish directly affect the search space and convergence rate of the algorithm. In order to increase the global searching ability and speed up the convergence, the larger visual field and step size can be selected at the earlier stage of the algorithm. In the later stage of the algorithm, the visual field and step size can be reduced. And the local search ability can be enhanced, so that the solution of the oscillation can be avoided and the solution precision of the algorithm can be improved. The corresponding adjustment rules are as follows.

$$\begin{cases} Visual = Visual \times exp\left(-30 \times \left(\frac{t}{T}\right)^{\mu}\right) + Visual_{min}; \\ Step = Step \times \exp\left(-30 \times \left(\frac{t}{T}\right)^{\mu}\right) + Step_{min}; \\ X_{next} = X + \frac{X_n - X}{\|X_n - X\|} \cdot \left|1 - \frac{Y}{Y_n}\right| Step \end{cases} \tag{4}$$

Among them, t is the current iteration number, and T is the maximum iteration number, $\mu \in [1,10]$, X_{next} is the movement rule of artificial fish under the current state of X and the next state X_n of search. To balance the local search and global search of the algorithm, setting *Visual* and *Step* as 3 piecewise functions, and in the early stage, the values are large and gradually become smaller, and finally maintain at $Visual_{min}$ and $Step_{min}$.

(2) Improvement of movement strategy. The random moving behavior of artificial fish can lead to the uncertainty of search and the degradation of the solution. To this end, bulletin board is introduced in the optimization process. And it always keeps to the direction of the better solution, and accelerate the convergence speed and accuracy of the algorithm and prevent the degradation of the optimal solution in the population [9]. Meanwhile, biological competition mechanism is introduced and death operator $\Gamma_d(X \in R^n, \Gamma_m(X) = rand() * (\mathrm{H} - \mathrm{L}) + \mathrm{L}$ is adopted, the upper and lower bounds of variables are H and L respectively. During the operation of the algorithm, some individuals whose target function values are smaller are eliminated. Then they are reinitialized to maintain the diversity of the population and enhance the global optimization ability of the algorithm.

2.2 Clustering Analysis Based on the Improved AFSA

Based on the user-item rating matrix and item-tag weight matrix, each user is treated as an artificial fish, we establish the objective function of artificial fish clustering by using Pearson Correlation Coefficient [1]. Through four kinds of behaviors of artificial fish to reach the optimization goal, and finally complete the analysis process of user clustering.

In this paper, we integrate the initial user clustering and user-item rating matrix to determine the different user clusters' preference items. Furthermore, the tag clustering can be obtained. Then, the initial clustering analysis of elements in the social tagging system is completed. And the dense initial analysis data set could be obtained, which will significantly reduce the data redundancy and data size. In general, the traditional K-means clustering algorithm is significantly affected by the initial clustering center and the minimum distance between cluster centers. Compared with this method, the improved AFSW has the characteristics of insensitivity to initial values and parameter values. Apart from simpler algorithm, easier implementation and stronger robustness, it also has better global search performance [8]. The implementation steps of the algorithm are as follows:

(1) Data preprocessing: Data cleaning, data transformation and normalization are processed for the initial data set, and establish user' vector space model. For instance, user sets: $U\{u_i | u_i = \left(u_{i1}, u_{i2}, \cdots, u_{in}\right), i = 1, 2, \cdots, m\}$, which denotes the users' data sets with $m*n$, u_{ij} is the j eigenvalue of the user i, u_i represents the individual state of the artificial fish. And basic parameters of the algorithm are set, such as δ, *Try_number*, initial cluster number k, and randomly select k values to determine the initial clustering center.

(2) Establishing objective function: The objective function $f(u)$ is the distance closeness function between u_i and cluster center z_j (food concentration), which is defined as follows:

$$\min f(u) = \sum\nolimits_{u_i \in Z_j} \sum\nolimits_{j=1}^{k} \left(sim\left(u_i, z_j\right)\theta + \frac{1}{D\left(u_i, z_j\right)}(1 - \theta) \right) \quad (5)$$

$sim(u_i, z_j)$ is the similarity between the user and the clustering center determined by the Pearson correlation coefficient, $D(u_i, z_j)$ is defined by Euclidean distance, θ. is the accommodation factor.

(3) Calculate the initial food concentration of artificial fish and give the initial value to the bulletin board. At the same time, the billboard update strategy in Sect. 2.1 is used to optimize the position information of the artificial fish.
(4) Compare the current food concentration of the artificial fish and the food concentration in the optimal partner area within the visual field. According to Formula (4), the visual field and step length of the adaptive artificial fish are calculated. And perform preying, swarming, following and other behaviors are compare with the initial clustering center state in order to update the status of the bulletin board or to generate the current optimal clustering center.
(5) Whether the maximum iteration number is reached or the termination condition is satisfied. If the condition is satisfied, the optimal solution is output. Otherwise, the (1) is transferred until the condition is satisfied.

3 Tensor Model and Recommendation Algorithm

3.1 Tensor and Tensor Decomposition

In algebraic geometry, a tensor is defined by a multiple linear function coefficient, which is delimited in the Cartesian product of the vector space and the duality space. Generally, the zero-order tensor is a scalar, the first-order tensor is similar to the vector, the second-order tensor is similar to the matrix, and the third-order tensor is similar to the cube matrix. In the tag system recommendation, users, items, and tags can be used as a third-order tensor to describe the elemental and ternary relations in the tag recommendation system.

The tensor decomposition is an approach to reduce the order of the high-order tensors. The purpose is to obtain a denser approximation tensor that is smaller than the original tensor, and to maximize the retention of the original tensor. In this way, a personalized recommendation service can be provided. CP decomposition and Tucker decomposition are the most common tensor decomposition methods [11]. CP decomposition is the decomposition of tensor into a finite number of orders, which is also the promotion of traditional matrix decomposition. Tucker decomposition is approximately the mode-n product of a kernel tensor G and a factor matrix. The formal definition of a Tucker decomposition model of the third order tensor $\mathrm{Y} \in \mathrm{R}^{\mathrm{I} \times \mathrm{J} \times \mathrm{K}}$ formed by a user, a item and a tag is defined as follows:

$$\mathrm{Y} \approx \hat{\mathrm{Y}} = G_{\times 1} U_{\times 2} V_{\times 3} W, \quad r_{\mathrm{ijk}} = \sum\nolimits_{\mathrm{p}=1}^{P} \sum\nolimits_{q=1}^{Q} \sum\nolimits_{R=1}^{R} u_{ip} v_{jq} w_{kr} g_{pqr} \tag{6}$$

Among them, $\mathrm{Y} \in \mathrm{R}^{\mathrm{I} \times \mathrm{J} \times \mathrm{K}}$ is the core tensor,$\mathrm{Y} \in \mathrm{R}^{\mathrm{I} \times \mathrm{J} \times \mathrm{K}}$, $\mathrm{V} \in \mathrm{R}^{\mathrm{Q} \times \mathrm{J}}$, $\mathrm{W} \in \mathrm{R}^{\mathrm{R} \times \mathrm{K}}$ is feature matrix on three dimensions. The order of the tensor Y is (P, Q, R), $\times_{\mathrm{i}}$ is the mode-i tensor-matrix multiplication, $\mathrm{r}_{\mathrm{ijk}}$ is the (i, j, k) element of the tensor $\hat{\mathrm{Y}}$. The principle of Tucker decomposition is shown in Fig. 1. The detailed steps of the tensor decomposition algorithm can be found in [12].

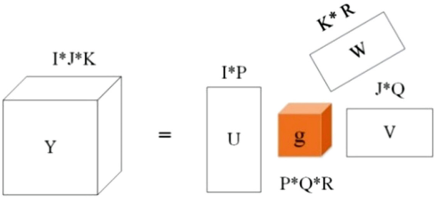


Fig. 1. Tucker decomposition principle

3.2 Tensor Decomposition with Dynamic Incremental Updating

The application of the Tucker decomposition model in the tag recommendation system exhibits good performance. However, with the new users and new items continuing to join, the traditional tensor decomposition algorithm based on the recommendation system needs resolving the tensor to repeat the operation. As such, it greatly increases the consumption of computing resources. To solve this problem, a dynamic increment of the tensor decomposition method can be used [13]. On the basis of the original tensor decomposition, with the new user and the new tensor constituted by item, tensor decomposition algorithm is carried out, and the original tensor decomposition results are dynamically updated, which greatly reduce the complexity of the algorithm and improve the adaptability of the algorithm.

The specific idea is: letting the current tensor Y, the new added tensor Y', then Y* = (Y + Y'). For the old users, the original decomposition of the results can be used to recommend. And for the new users, it only needs to update the tensor decomposition model part of the parameters and get the prediction value of Y'. And then the recommendation process can be completed, without repeating decomposition calculation of the entire tensor Y*. The specific solution is as follows:

Set data $\mathrm{X}^{\mathrm{old}}(\mathrm{X}_1, \mathrm{X}_2, \cdots, \mathrm{X}_{\mathrm{N}_\mathrm{k}})$, new user data $\mathrm{X}^{\mathrm{new}}(\mathrm{X}_{\mathrm{N}_{\mathrm{k+1}}}, \mathrm{X}_{\mathrm{N}_{\mathrm{k+2}}}, \cdots, \mathrm{X}_{\mathrm{N}_{\mathrm{k+l}}})$, and W turn from $N_k \times R$ to $(N_k + l) \times R$, l is the number of added users, W is decomposed into $W = (\mathrm{W}^{\mathrm{old}}, \mathrm{W}^{\mathrm{new}})$, where $\mathrm{W}^{\mathrm{old}}$ remains the same, then only $\mathrm{W}^{\mathrm{new}}$ needs to be solved. The corresponding objective function is: $\mathrm{Min\,J} = \left\|\mathrm{X}^{\mathrm{new}} - \mathrm{G} \times_1 \mathrm{U} \times_2 \mathrm{V} \times_3 \mathrm{W}^{\mathrm{new}}\right\|$, keep ($G$, U, V) unchanged, then the optimal solution of $\mathrm{W}^{\mathrm{new}}$ is $(\mathrm{X}^{\mathrm{new}})^{\mathrm{T}}\mathrm{A}\left(\mathrm{A}^{\mathrm{T}}\mathrm{A}\right)^{-1}$, the optimal solution of $\mathrm{Y}^{\mathrm{new}} = \sum_{\mathrm{k}=1}^{\mathrm{l}} \sum_{\mathrm{r}=1}^{\mathrm{R}} \mathrm{U}\mathrm{G}^{\mathrm{r}}\mathrm{V}^{\mathrm{T}}(\mathrm{W}^{\mathrm{new}})$.

3.3 Tensor Modeling and Recommendation Algorithm

In the algorithm of socialized tag recommendation, the tensor model is established according to the ternary relation of the "user-item-tag". The tensor representation and modeling methods are mainly two forms: the positive "0/1" and the negative "±" [14]. Although this approach is easy to understand and deal with, the importance of the elements of the tag system is overlooked. For this reason, based on the "user-item-tag" relationship, we propose to consider the weight of the elements and the score information which reflects the user's preference. The weighting of the elements and the score attributes as the elements in the tensor. The new weighted tensor model is used to solve the model by using the tensor decomposition algorithm based on the incremental update proposed in Sect. 3.2, and the data preprocessing is processed by the method proposed in Sect. 2.

For the user (i), item (j) and tag (k) as a triplet relationship, the attributes of its tag are extracted. The Euler distance method is used to define the relationship between the corresponding items $\mathrm{D}\left(\mathrm{t}_{\mathrm{k}}, \mathrm{t}_{\mathrm{c}}\right)$. Then the $\mathrm{w}_{\mathrm{ijk}} = 1/\mathrm{D}\left(\mathrm{t}_{\mathrm{k}}, \mathrm{t}_{\mathrm{c}}\right)$ is defined as the weight value of the tag in the triplet, and R_{ij} is defined as the user i for the item j (0–5), we get the weight $\mathrm{W}_{\mathrm{ijk}} = \mathrm{R}_{\mathrm{ij}} * \mathrm{w}_{\mathrm{ijk}}$ (user-item-tag). And then we use the triplet weight $\mathrm{W}_{\mathrm{ijk}}$ as the tensor element to establish the tensor model. In Sect. 3.2, the incremental updating of the tensor decomposition algorithm is used to solve the model. Finally, based on the size of the weight, the optimal TOP-N tag recommendation list is generated for a user on the item. In the user recommendation and item recommendation application, similar methods can be used.

4 Experiment Design and Data Analysis

We selected two different testing datasets as the samples of this study. Then, the proposed FTA algorithm is compared with two typical tag recommendation algorithms i.e. FolkRank and Tensor Decomposition in terms of recommendation performance. To test the validity of our model, Precious-Recall indices were adopted as the evaluation methods for the recommendation quality.

4.1 Testing Datasets

The testing datasets used in this study were downloaded from two open-source datasets websites, including MovieLens and Last.fm, which provided the empirical datasets for testing the performance of recommendation algorithm [15, 16]. The detailed sample scale and characteristics were shown in Table 1.

Table 1. Testing datasets scale and characteristics

Data Sets	Users	Items	Tags	Ratings
MovieLens	525	834	2804	569
Last.fm	756	728	3183	405

4.2 Algorithm Evaluation Indices

Previously, Precious–Recall ratios were defined as: precision ratio reflects the proportion of the number of users' actual preference labels among the label datasets in the system; recall ratio represents the percentage of users whose preference labels were included in the final recommendation list. In our study, we made some modifications to the former definition: Assuming that $P(u)$ represents the testing user's preference tag waiting for prediction, $T(u)$ denotes the *TOP-N* recommendation list provided by the system, $R(u)$ refers to the user's actual preference tag during the system interaction process, then $Precision = T(u) \cap P(u)\,/N, Recall = \ T(u) \cap R(u)\,/P(u) \cap R(u).$

4.3 Experiment Results and Discussions

Considering the consistence of performance comparison, we select *TOP-N = [6, 7, 8, 9, 10],* experiments in each scenario were conducted for 10 repetitions, respectively. Then, the optimal results were averaged as the final evaluation results. As such, we can control the influence of data distribution and parameter adjustments on the performance of algorithm. The experiment results of two different datasets were displayed in Figs. 2 and 3, respectively.

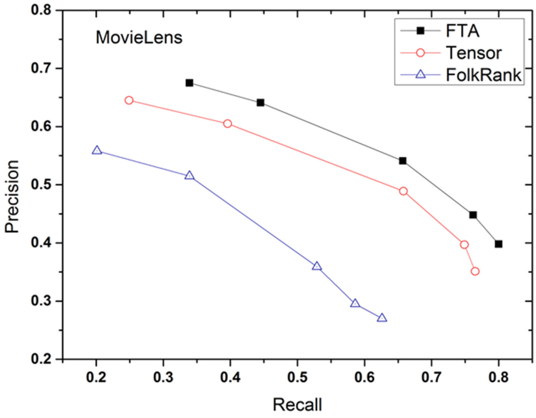


Fig. 2. Comparison results of different algorithms on MovieLens datasets

Results show that, the FTA algorithm proposed in this paper exhibits adaptability in terms of the performance in both the two datasets. The quality of recommendation is better than the two traditional algorithms. Moreover, from the results, we can make the following observations:

(1) With the increase of *TOP-N*, Precision and Recall curves display a contradictory trend. At the lower value of *N*, the Precision of the results is larger than the Recall and vise verse. This is arising from the nature of Precision-Recall indices. However, with regards to different algorithms, the two indices have different performance.

(2) Traditional tensor decomposition algorithm can have good performance under normal situations. However, in terms of large scale and sparse dataset, it often fails to control the complexity. Anyway, it still has better performance than FolkRank method.

(3) FTA algorithm is an effectively tool for dimension reduction. Meanwhile, the adoption of tensor decomposition strategy with incremental updating can shorten the processing time and preserve the latent information among the 3D elements. Also, we can achieve better recommendation quality compared with the former two algorithms.

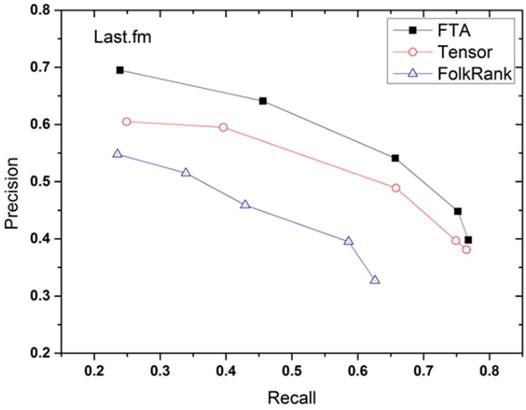


Fig. 3. Comparison results of different algorithms on Last.fm datasets

5 Conclusion

The wide use of Socialized Tag System (STS) provides a boarder information platform to the users. Consequently, it brings the problems of information overloading. Conventional recommendation systems provide personalized recommendation service on the basis of the pairwise relations of user and item i.e. “user-user”, “user-item” and “item-item”. However, such system has limitations with respect to dealing with the 3D relations (“user-item-tag”) in STS. This study integrates a modified Artificial Swarm Fish Algorithm (ASFA) and tensor modeling approach to establish a new type of STS recommendation model. Firstly, we reduce the dimension of our dataset through clustering the elements of the STS using our modified ASFA. Then, we apply a tensor decomposition algorithm with incremental updating to model the recommendation system. This step enables us to maximally keep the latent information in STS while still satisfies the required performance. Finally, we utilize two testing empirical datasets to validate the proposed algorithm and compare the results with two conventional algorithms. Results indicate that our method has better performance in terms of both precision and recall ratios. However, in our research process, several problems are identified and remained to be solved. For instance, the complexity of the algorithm can be further reduced. And more emulation methods should be adopted. The above two issues are regarded as the future research directions.

Acknowledgements. This paper is funded by natural science fund for colleges and universities in Jiangsu Province(17KJB580001) and philosophy and social fund for colleges and universities in Jiangsu Province(2017SJB1641) and Huai'an municipal science and technology bureau project(HAR201608) and Huai'an college of information technology project.

References

1. Lü, L., Medo, M., Yeung, C.H., et al.: Recommender systems. Phys. Rep. **519**, 1–49 (2012). https://doi.org/10.1016/j.physrep.2012.02.006
2. Chao, C., Ying-chao, Z., Jin, M.: A collaborative filtering recommender algorithm based on tripartite network. J. Nanjing Univ. Inf. Sci. Technol. Nat. Sci. Ed. **2**, 337–339 (2010)
3. Liao, Z.F., Li, L., Liu, L.M., Li, Y.Z.: A tripartite decomposition of tensor for social tagging. Chin. J. Comput. **35**, 2625 (2012)
4. Symeonidis, P., Nanopoulos, A., Manolopoulos, Y.: A unified framework for providing recommendations in social tagging systems based on ternary semantic analysis. IEEE Trans. Knowl. Data Eng. **22**, 179–192 (2009)
5. Sun, L., Li, S.: Social tagging recommendation system based on K-means cluster and tensor decomposition. J. Jiangsu Univ. Sci. Technol. **7**, 8 (2012)
6. Long, W., Jialun, W., Zhuanli, C., et al.: Personalized medicine recommendation based on tensor decomposition. Comput. Sci. (2015). https://doi.org/10.11896/j.issn.1002-137X.2015.5.045
7. Jabeen, F., Khusro, S., Majid, A., Rauf, A.: Semantics discovery in social tagging systems: a review. Multimed. Tools Appl. **75**, 573–605 (2016)
8. Nieming, G.: Artificial Fish Swarm Algorithm and its Applications. Guangxi University for Nationalities, Nanning (2009)
9. Peng, Y., Tang, G.L., Xue, Z.C.: Optimal operation of cascade reservoirs based on improved artificial fish swarm algorithm. Syst. Eng. Pract. **31**, 1118–1125 (2011)
10. Liao, Y., Peng, L., Jian, W., Zhang, M.: Control parameter optimization for the unmanned surface vehicle with the improved artificial fish swarm algorithm. J. Harbin Eng. Univ. **35**, 800–806 (2014)
11. Weng, S.S., Lin, B., Chen, W.T.: Using contextual information and multidimensional approach for recommendation. Expert Syst. Appl. Int. J. **36**(2), 1268–1279 (2009)
12. Gui, L.I., Wang, S., Zheng-Yu, L.I., et al.: Personalized tag recommendation algorithm based on tensor decomposition. Comput. Sci. (2015)
13. Zou, B.Y., Cui-Ping, L.I., Tan, L.W., et al.: Social recommendations based on user trust and tensor factorization. J. Softw. (2014)
14. Milicevic, A.K., Nanopoulos, A., Ivanovic, M.: Social tagging in recommender systems: a survey of the state-of-the-art and possible extensions. Artif. Intell. Rev. **33**, 187–209 (2010)
15. In MovieLens. www.movielens.org. Accessed Jan 2017
16. In LastFM. www.last.fm. Accessed Jan 2017

Research on the Nearest Neighbor Representation Classification Algorithm in Feature Space

Yan-Hong Hu[1], Yu-Hai Li[2], and Ming Zhao[3](✉)

[1] School of Occupational & Continuing Education, Central China Normal University,
Wuhan 430070, People's Republic of China
[2] School of Information Management, Central China Normal University, Wuhan 430070,
People's Republic of China
[3] School of Computer Technology, Yangtze University, Jinzhou 434023,
Hubei, People's Republic of China
hitmzhao@gmail.com

Abstract. Representation-based classification and recognition, such as face recognition, have dominant performance in dealing with high-dimension data. However, for low-dimension data the classification results are not satisfying. This paper proposes a classification method based on nearest neighbor representation in feature space, which extends representation-based classification to nonlinear feature space, and also remedies its drawback in low-dimension data processing. First of all, the proposed method projects the data into a high-dimension space through a kernel function. Then, the test sample is represented by the linear combination of all training samples and the corresponding coefficients of each training sample will be obtained. Finally, the test sample is assigned to the class of the training sample with a minimum distance. The results of experiments on standard two-class datasets and ORL and YALE face databases show that the algorithm has better classification performance.

Keywords: Nearest neighbor classification · Representation · Kernel function

1 Introduction

The nearest neighbor classifier [1], as one of the classical classifiers, has been long drawing people's attention. The principle by which it works is to assign the pattern to the class of the sample that is the closest to it. The performance of the nearest neighbor classifier depends on the method of distance calculation. The similarity between two patterns is usually measured by distance functions. That is to say, when classifying a new pattern, we should first identify the pattern closest to the test pattern, which then enables the new pattern to be assigned to the class of the training pattern closest to it. In order to improve the performance of the nearest neighbor classifier, the distribution characteristics of various patterns in the subspace needs to be enhanced. On this basis, Li *et al.* [2] put forward the nearest feature line (NFL) method. And moreover, Chien *et al.* [3] proposed the nearest feature plane (NFP) method. Both NFL and NFP have

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 14–19, 2018.
https://doi.org/10.1007/978-3-319-76451-1_2

improved the classification ability of the nearest neighbor classifier. Obviously however, they increased its time complexity.

Extensive attention has been paid to find out a suitable measure to identify the nearest neighbor of a sample in recent years. Many methods were proposed for the effective search of the nearest neighbor. For example, Samet [4] proposed the maximum nearest neighbor algorithm to find out the k-nearest neighbor; Wang *et al.* [5] proposed a simple method of adaptive distance test to mark the different weights of each sample. What needs to be noted, however, is that these algorithms have ignored the relationships among different samples. In other words, these algorithms calculate the distance between the test sample and the training sample without taking into consideration the connection among the training samples, which could lead to classification errors.

Some other measures are used to calculate the distance between the test sample and the training sample in other methods, such as kernel classification [6, 7], sparse-representation based classification [8], and the like. The former classifies through mapping the samples to a high-dimension space via nonlinear mapping. The latter represents test samples by means of the weight sum of training samples. However, when classifying low-dimension data, representation based on classification methods will be influenced by dimensionality. Therefore, in order to improve the quality of low-dimension data classification, this paper proposes a nearest-neighbor-representation based classification in feature space, where the data are first projected into a high-dimension space and then represented and classified. Experimental results prove that this method has good classification performance.

2 Representation Based Classification

Sparse-representation can be used to find the sparsest set of coefficients to represent test samples by solving the following optimization problems:

$$\min_{x} \|x\|_1 \ \textit{subject to } Ax = y \tag{1}$$

Here A is the matrix of training samples, while $\|\bullet\|_1$ is the norm l^0, and x is the coefficient of y.

After obtaining the solution of formula (1), sparse-representation assigns the test samples to the class which has least redundant errors through reconstruction. The shortcoming of this method is that it costs too much time working out the sparse coefficients.

3 The Classification of the Nearest Neighbor Representation in Feature Space

Assuming that in the original space, there are n training samples $x_1, x_2, \ldots, x_n$ which fall into c types, and that y is a test sample in the original space. Our method first projects the test sample y into the feature space, then it is represented and classified by the linear combination of training samples. The feature space is obtained by projecting the original space through nonlinear mapping ϕ. The test sample y becomes $\phi(y)$ when it is projected into the feature space, and training samples become $\phi(x_1), \phi(x_2), \ldots, \phi(x_n)$. Therefore, we can get the formula $\phi(y) = \sum_{i=1}^{n} \alpha_i \phi(x_i)$. Each column vector in the feature space stands for a sample. The formula $\phi(y) = \sum_{i=1}^{n} \alpha_i \phi(x_i)$ can be rewritten as:

$$\phi(y) = \Phi\Psi \tag{2}$$

Here $\Phi = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)], \Psi = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$. We cannot directly work out the solution of formula (2), since Φ is unknown. But it can be converted to the following formula:

$$\Phi^T \phi(y) = \Phi^T \Phi\Psi \tag{3}$$

According to the kernel function defined in reference [9], formula (3) can be converted to

$$K_y = K\Psi \tag{4}$$

Here $K_y = \begin{pmatrix} k(x_1, y) \\ \vdots \\ k(x_n, y) \end{pmatrix}, K = \begin{pmatrix} k(x_1, x_1) \cdots k(x_1, x_n) \\ \vdots \\ k(x_n, x_1) \cdots k(x_n, x_n) \end{pmatrix}, \Psi = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}$.

If K is non-singular square matrix, we can directly get the solution of formula (4), namely $\Psi = K^{-1}K_y$; otherwise, we can use the formula $\Psi = (K + \mu I)^{-1}K_y$, where μ stands for a positive constant and I stands for an identity matrix, to solve formula (4).

4 Numerical Experiments

In order to verify the validity of our algorithm, we applied it to test two-class databases, ORL [10] and YALE [11] face databases. Gaussian kernel $k(x, y) = \exp(-\|x - y\|^2/2\sigma)$, in which σ is a parameter, is chosen as the kernel function in this paper.

4.1 Results of Experiments on Standard Two-Class Datasets

The experiment shows that our algorithm is very suitable for low-dimension data classification. Table 1 gives information of the dimensionality of each dataset and the size of the samples. It can be observed that the dimensionalities of these datasets are very low. If the classification quality is not satisfactory in the original space, it will be significantly improved by projecting the original data into a high-dimension space through nonlinear function ϕ. Table 2 shows that our method has considerably higher classification accuracy than INNC [12] and LRC [13].

Table 1. Two-class datasets

Datasets	Number of training datasets	Number of test datasets	Dimensionality
Banana	400	4900	2
Heart	170	100	13
German	700	300	20
Image	1300	1010	18
Thyroid	140	75	5
Diabetes	468	300	8
Titanic	150	2051	3

Table 2. Classification accuracy (mean value ± standard deviation)

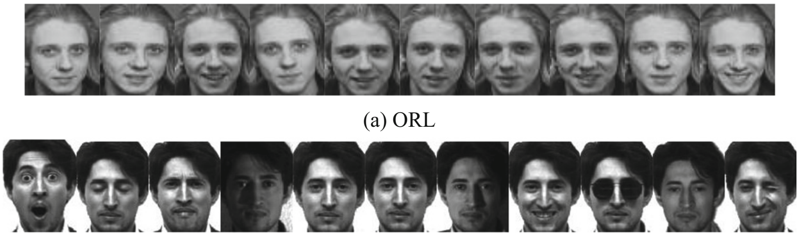
Dataset	Our method	INNC	LRC
Banana	83.87 ± 0.74	51.77 ± 5.97	60.45 ± 2.20
Heart	81.53 ± 2.93	61.37 ± 4.69	61.31 ± 5.22
German	72.03 ± 1.80	63.74 ± 2.72	70.21 ± 2.04
Image	95.71 ± 0.39	75.52 ± 1.54	74.59 ± 2.48
Thyroid	94.16 ± 1.69	29.96 ± 4.42	29.79 ± 4.45
Diabetes	74.57 ± 2.00	52.01 ± 3.53	64.47 ± 2.57
Titanic	65.75 ± 1.31	45.11 ± 9.10	63.76 ± 2.73

4.2 Results of Experiment on Face Databases

This part presents the results of the experiment that apply our method to ORL and YALE face databases. The results are compared with those of experiments respectively using nearest neighbor line (NNL) [14], center-based nearest neighbor classifier (CBNN) [15], nearest neighbor classifier (NNC) [1] INNC and large-margin nearest neighbor classifiers via sample weight learning (L-SWL) [16].

ORL face database contains 400 face photos of 40 different persons, 10 for each. The photos of each person are taken in different time and under different light conditions, and they show different facial expressions. In our experiment, each photo is cut into the size of 46 × 56 pixels. YALE face database contains face photos of 11 persons, 15 for each. These photos show different facial expressions, such as normal, sad, happy, amazed, and so on. Figure 1 shows the photos of a person in ORL (a) and YALE (b), respectively.

In the experiment on ORL face database, two evaluation programs are used. Program 1 is to select the first six photos of each person as the training samples, and the remaining are test samples. Program 2 is to select 5 photos of each person as training samples, and the others are test samples. By the same token, two evaluation programs are used in YALE face database experiment. Program 1 is to select the first six photos of each person as training samples, and the remaining are test samples; program 2 is to select the first 7 photos of each person as training samples, and the left as test samples.



(a) ORL

(b) YALE

Fig. 1. Sample face photos of ORL and YALE face databases

Tables 3 and 4 show that our method is better than others. In particular, Table 4 indicates that our method has considerably improved classification accuracy, while CBNN, NNL, NNC and L-SWL have the same relatively lower classification accuracy. In evaluation programs 1 and 2, the classification accuracy of our method is 2.22% and 3.34% higher, respectively.

Table 3. Classification accuracy in ORL face database (μ = 0.5)

Evolution program	Our algorithm	INNC	CBNNC	NNL	NNC	L-SWL
Program 1	96.63% (1.0e6)	94.60%	93.13%	93.75%	95.00%	88.75%
Program 2	96.00% (1.0e7)	92.50%	88.50%	92.00%	91.50%	74.50%

Table 4. Classification accuracy in YALE face database (μ = 0.01)

Evolution program	Our algorithm	INNC	CBNNC	NNL	NNC	L-SWL
Program 1	96.78% (1.0e9)	95.56%	95.56%	95.56%	95.56%	95.56%
Program 2	97.67% (1.0e7)	86.67%	93.33%	93.33%	93.33%	93.33%

5 Conclusion

This paper proposes a nearest neighbor representation classification algorithm in feature space, which extends nearest neighbor classification to nonlinear space. When dealing with low-dimension data, traditional representation-based classification may undergo loss of classification performance. This paper solves this problem by making use of kernel function. In particular, our method has outstanding classification performance in two-type datasets.

References

1. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theor. **IT-13**(1), 21–27 (1967)
2. Li, S.Z., Lu, J.W.: Face recognition using the nearest feature line method. IEEE Trans. Neural Netw. **10**(2), 439–443 (1999)
3. Chien, J.T., Wu, C.C.: Discriminant waveletfaces and nearest feature classifiers for face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **24**(12), 1644–1649 (2002)
4. Samet, H.: K-nearest neighbor finding using MaxNearestDist. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 243–252 (2008)
5. Wang, J., Neskovic, P., Cooper, L.N.: Improving nearest neighbor rule with a simple adaptive distance measure. Pattern Recogn. Lett. **28**(2), 207–213 (2007)
6. Peng, Q.S., Wei, W.H.: Parallel fuzzy clustering algorithm based on kernel method. Comput. Eng. Des. **29**(8), 1881–1883 (2008)
7. Tao, D.C., Tang, X.O., Li, X.L., Rui, Y.: Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. IEEE Trans. Multimed. **8**(4), 716–727 (2006)
8. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **31**(2), 210–227 (2009)
9. Xu, Y., Zhu, Q., Chen, Y., Pan, J.S.: An improvement to the nearest neighbor classifier and face recognition experiments. Int. J. Innov. Comput. Inf. Control **9**(2), 543–554 (2013)
10. Samaria, F.S., Harter, A.C.: Parameterisation of a stochastic model for human face identification. In: Proceedings of the 2nd IEEE International Workshop on Applications of Computer Vision, Sarasota, FL, pp. 138–142, December 1994
11. Bellhumer, P.N., Hespanha, J., Kriegman, D.: Eigenfaces vs Fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7), 711–720 (1997)
12. Xu, Y., Fan, Z.Z., Zhu, Q.: Feature space-based human face image representation and recognition. Opt. Eng. **51**(1), 017205-1–017205-7 (2012)
13. Naseem, T.R., Bennamoun, M.: Linear regression for face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **32**(11), 2106–2112 (2010)
14. Zheng, W.S., Zhao, L., Zou, C.: Locally nearest neighbor classifiers for pattern classification. Pattern Recogn. **37**(6), 1307–1309 (2004)
15. Gao, Q.B., Wang, Z.Z.: Center-based nearest neighbor classifier. Pattern Recogn. **40**(1), 346–349 (2007)
16. Hu, Q.H., Zhu, P.F., Yang, Y.B., Yu, D.: Large-margin nearest neighbor classifiers via sample weight learning. Neurocomputing **74**(4), 656–660 (2011)

A New Aesthetic QR Code Algorithm Based on Salient Region Detection and SPBVM

Li Li[1], Yanyun Li[1], Bing Wang[1], Jianfeng Lu[1](✉), Shanqing Zhang[1],
Wenqiang Yuan[1], Saijiao Wang[2], and Chin-Chen Chang[3]

[1] Hangzhou Dianzi University, Hangzhou, China
jflu@hdu.edu.cn
[2] Taizhou Radio & TV University, Taizhou, China
[3] Feng Chia University, Taichung, Taiwan

Abstract. Many aesthetic QR code algorithms have been proposed. In this paper, a new aesthetic QR code algorithm, based on salient region detection and Selectable Positive Basis Vector Matrix (SPBVM), is proposed. Firstly, the complexity of texture features are added to calculate the saliency values, based on the existing salient region detection algorithm. According to the saliency map, the important area of the image is preserved for the subsequent beautification operation. Then, the appropriate basis vectors are selected by using the proposed SPBVM according to the acquired salient region, and the salient region is displayed completely by XOR operation which is performed by the generated original QR code and the selected basis vectors. Finally, the aesthetic QR code is obtained by combining the background image and the original QR code. The results show that the pro-posed algorithm can produce more accurate salient area and have more pleasant visual effect.

Keywords: Aesthetic QR code · RS code · Salient region detection
Selectable Positive Basis Vector Matrix · XOR operation

1 Introduction

QR code is a kind of two-dimensional code which has the characteristics of high error tolerance level, wide range of encoding information types as well as convenient and fast reading speed. The traditional QR code is composed of black and white modules, which all have single color and poor visual effect. Therefore, many researchers have focused on the algorithm of aesthetic QR code in order to solve the above problems.

The current aesthetic algorithms can be grouped into two categories. The first category modifies the color or shape of the module in the original QR code to achieve the beautification effect [1]. The results are shown in Fig. 1.

The algorithms in second category consider the fusion of the original QR code and background image to enhance the visual effects. Because the QR code is encoded by the coding mechanism of RS code, the algorithm can be divided into two kinds of algorithms according to the characteristics of RS code [2, 3].

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 20–32, 2018.
https://doi.org/10.1007/978-3-319-76451-1_3

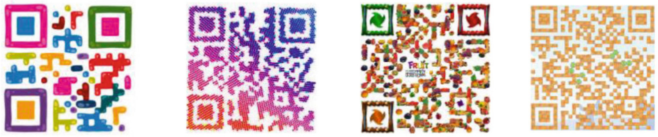
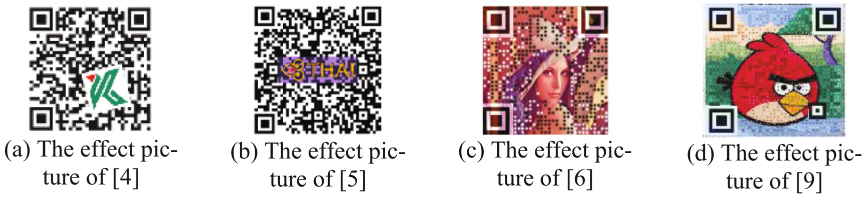


Fig. 1. Modifying the color or shape of the module.

The first kind of algorithms make use of the error correction coding mechanism of RS code [4–6, 9]. The effect pictures are shown in Fig. 2.



(a) The effect picture of [4]

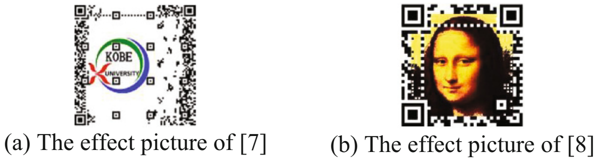
(b) The effect picture of [5]

(c) The effect picture of [6]

(d) The effect picture of [9]

Fig. 2. The aesthetic QR codes of different algorithms.

The second kind of algorithms uses the XOR characteristic of RS code [7, 8] to beautify the QR code. The effect pictures are shown in Fig. 3.



(a) The effect picture of [7]

(b) The effect picture of [8]

Fig. 3. The aesthetic QR codes of different algorithms.

Although the above algorithms have implemented the visual appearance of the traditional QR code. But the selected images for aesthetic QR code is relatively simple, the algorithms cannot be well applied to the background image with multiple salient regions.

Considering the above disadvantages, a new algorithm is produced as follows.

(1) A salient region detection algorithm is proposed to obtain a saliency map which is more suitable for the aesthetic QR code. The complexity of texture features is calculated by the Gray Level Co-occurrence Matrix (GLCM) and added into the salient region detection algorithm of Xu *et al.* [10], and the saliency map can be obtained more accurately.

(2) SPBVM is constructed according to the saliency map from Positive Basis Vector Matrix (PBVM), which can be used to realize the flexible replacement of regions.

Due to the basis vectors corresponding to the code words in data area are redundant in the process of the Gauss Jordan Elimination. Therefore, based on the saliency map, some positive basis vectors can be selected to form SPBVM, and the associated Reverse Basis Vector Matrix (RBVM) can be obtained by Gauss Jordan Elimination. Finally, the display of salient region is retained as much as possible.

2 The Proposed Algorithm of Aesthetic QR Code

Given a background image I_O, we intend to make the salient region of I_O in the aesthetic QR code I_R be fully displayed, and not affect the decoding rate. Firstly, the salient region of the selected background image I_O is obtained by the improved salient region detection algorithm. The complexity of texture features in background image is added to calculate the saliency values, and the saliency map I_{SAL} is generated. Then the binary image with closed area I_{BIN} is obtained by dilation and erosion algorithm. It is convenient for the selection of basis vectors. Secondly, the proposed SPBVM is used to generate the initial aesthetic QR code. According to the extracted salient region, the result of RBVM I_{RR} is generated by XOR operation between I_{BIN} and the selected basis vectors which include in the SPBVM. Finally, the aesthetic QR code I_R is generated by the synthetic strategy with I_{RR} and I_P. Figure 4 shows the flowchart of the proposed aesthetic algorithm.

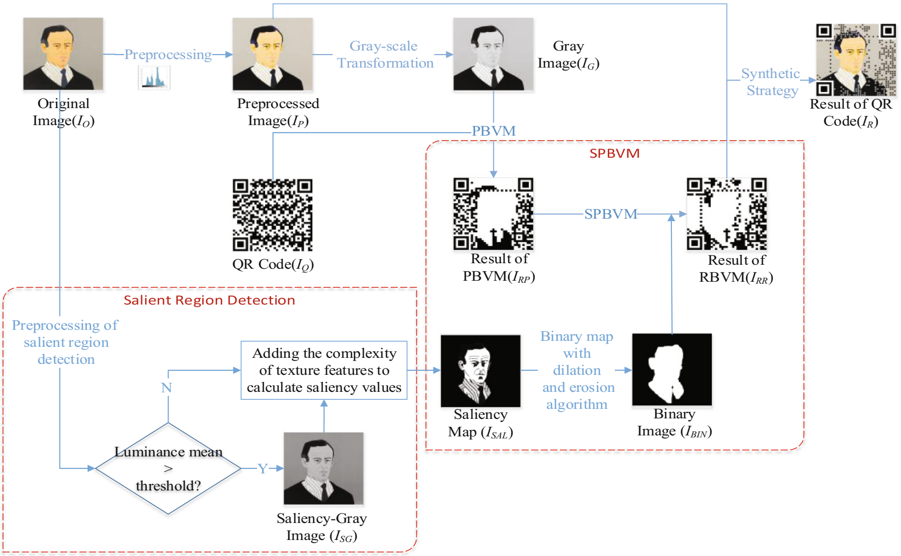

Original Image(I_O)
Preprocessing
Preprocessed Image(I_P)
Gray-scale Transformation
Gray Image(I_G)
Synthetic Strategy
Result of QR Code(I_R)
PBVM
SPBVM
QR Code(I_Q)
Result of PBVM(I_{RP})
SPBVM
Result of RBVM(I_{RR})
Salient Region Detection
Preprocessing of salient region detection
N
Adding the complexity of texture features to calculate saliency values
Luminance mean > threshold?
Y
Saliency-Gray Image (I_{SG})
Saliency Map (I_{SAL})
Binary map with dilation and erosion algorithm
Binary Image (I_{BIN})

Fig. 4. Flowchart of the proposed algorithm.

2.1 The Improved Salient Region Detection Algorithm

In the previous works, without taking into account of the salient region of the background image, there are black and white modules in the salient area of the final results. Therefore, the salient region detection algorithm is added to the aesthetic QR code algorithm in this paper.

In this paper, the salient region detection algorithm is improved on the basis of Xu *et al.* [10] and Cheng *et al.* [11]. In the previous algorithm, the brightness variation of the elements is taken into account. However, it ignores the situation where the brightness of the non-salient region changes greatly, the non-salient region is mislabeled as a salient region, resulting in inaccurate detection results. Considering that the salient region has richer texture, the texture in non-salient region is however relatively simple and smooth. Therefore, by adding the complexity of texture features in each region, the more accurate salient region can be achieved.

The description of the complexity of texture features. The Gray Level Co-occurrence Matrix (GLCM) [12–15] is used to calculate the complexity of texture features. Five texture parameters (energy, entropy, contrast, homogeneity, correlation) are selected in the salient region detection algorithm, and the saliency map can be applied to the aesthetic QR code. The complexity of texture features is calculated the Eq. (1).

$$\begin{aligned} G_{R_k}^{com} = {} & a_1 * Energy + a_2 * Entropy + a_3 * Contrast + a_4 * Homogeneity \\ & + a_5 * Correlation \end{aligned} \tag{1}$$

In the Eq. (1), a_1, a_2, a_3, a_4 and a_5 denote the weight coefficients, which are calculated based on the BP neural network in Chen *et al.* [16]. *Energy* represents the distribution of grayscale image and the thickness of the texture, in Eq. (2). *Entropy* represents the amount of information in the image, in Eq. (3). *Contrast* reflects the comparison of the brightness between a pixel value and its neighborhood pixel values, in Eq. (4). *Homogeneity* represents the local changes of the texture, in Eq. (5). *Correlation* represents the conformity of the texture in different directions, in Eqs. (6–10).

$$Energy = \sum_{i=1}^{n}\sum_{j=1}^{n} G(i,j)^2. \tag{2}$$

$$Entropy = -\sum_{i=1}^{n}\sum_{j=1}^{n} G(i,j)\log_2 G(i,j). \tag{3}$$

$$Contrast = \sum_{i=1}^{n}\sum_{j=1}^{n} (i-j)^2 G(i,j). \tag{4}$$

$$Homogeneity = \sum_{i=1}^{n}\sum_{j=1}^{n} 1/(1+(i-j)^2) * G(i,j). \tag{5}$$

$$Correlation = \sum_{i=1}^{n}\sum_{j=1}^{n}((i * j)G(i,j) - u_1u_2)/s_1s_2. \tag{6}$$

Where

$$\mathrm{u}_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} i * G(i,j). \tag{7}$$

$$\mathrm{u}_2 = \sum_{i=1}^{n}\sum_{j=1}^{n} j * G(i,j). \tag{8}$$

$$s_1^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} G(i,j) * (i - u_1)^2. \tag{9}$$

$$s_2^2 = \sum_{i=1}^{n}\sum_{j=1}^{n} G(i,j) * (i - u_2)^2. \tag{10}$$

The flowchart of salient region detection algorithm. The improved salient region detection algorithm is described in details as follows:

Step 1: The background image has to be preprocessed. The luminance mean of the image is calculated to compare with the fixed threshold. If the luminance mean is greater than the threshold, the image will be converted to grayscale image.
Step 2: The image is divided into different regions by the segmentation algorithm.
Step 3: The complexity of texture features is calculated as a detection factor of the salient region.
Step 4: The salient values are mapped to each pixel of the corresponding region.
Step 5: The binary map of salient region is obtained by Otsu's method [17], and then the dilation and erosion algorithms are used to get a closed area.

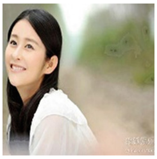
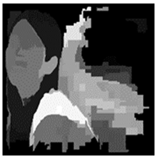
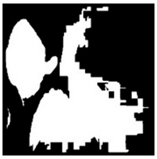
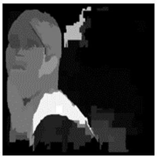
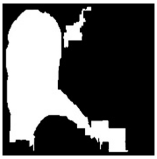
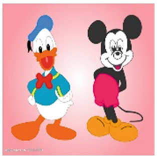
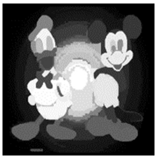
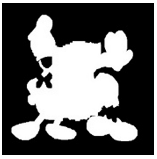
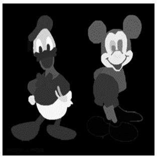
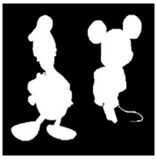
The saliency values are calculated by the Eq. (11).

$$S(R_k) = G_{R_k}^{com} L_{R_k} \sum\nolimits_{r_k \neq r_i} e^{D_s(R_k,R_i)/-\sigma_s^2} w(R_i) D_R(R_k, R_i). \tag{11}$$

Here, $G_{R_k}^{com}$ is the complexity of texture features. L_{R_k} is the brightness variation of the elements. $D_s(R_k, R_i)$ indicates the spatial distance between different regions. $w(R_i)$ represents the weight of the region, determined by the number of pixels. $D_R(R_k, R_i)$ represents the color distance between the regions.

The comparison between the improved salient region detection algorithm and Xu's algorithm in [10] is shown in Table 1.

Table 1. The comparison of different salient region detection algorithms

Background image	Saliency map of [10]	Binary image of [10]	Saliency map of our algorithm	Binary image of our algorithm
				
				

2.2 The Generation of SPBVM

In order to display the salient region as much as possible, we propose the concept of SPBVM.

Distribution and characteristics of RS code. The RS code can be divided into two parts: data area (the length is k) and parity area (the length is t), the total length of a RS code is n. Among them, the area for storing the input information is called as valid data area (the length is m), and a terminator is added after the information sequence, the code words after the terminator is called as invalid data area (the length is k-m). The information stored in the invalid data area has no effect on decoding, so it is often modified to display the background image in the algorithm of aesthetic QR code.

The distribution of RS code in QR code is shown in Fig. 5.

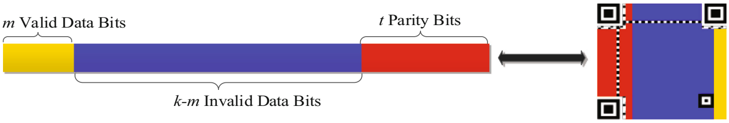


Fig. 5. The distribution of RS code in QR code.

In [2, 3], Cox *et al*. have proved that the RS code has two important characteristics. (1) The input information can be obtained directly in the encoding procedure; (2) The XOR operation can be performed in RS code. That is, two different RS codes are transformed into a new RS code by XOR operation.

PBVM and RBVM. RS encoding is closed under XOR and the concept of basis matrix are proposed in [2, 3]. The modification of data area is achieved by the closed of RS encoding. The idea is improved by introducing the optimization fusion strategy in [8]. We extend the idea and propose the data area is modified by PBVM and RBVM is used to modify the parity area.

First of all, the positive basis vector b_i is constructed with the former k data bits and the latter t parity bits, only the i_{th} bit is 1 in the data bits, the other bits are 0, and the t bits are the corresponding parity bits of RS code. Each basis vector is a valid RS code where $1 \le i \le k$. The all positive basis vectors b_i are represented in Fig. 6.

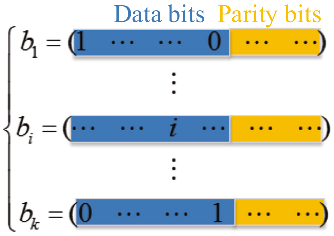

Data bits
Parity bits
$b_1 = (1 \cdots \cdots 0 \cdots \cdots)$
$b_i = (\cdots \cdots i \cdots \cdots \cdots)$
$b_k = (0 \cdots \cdots 1 \cdots \cdots)$

Fig. 6. The positive basis vector b_i.

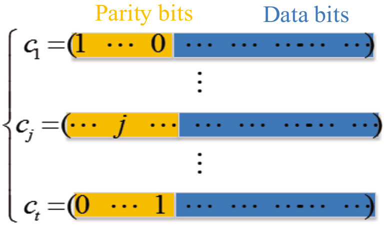

Parity bits
Data bits
$c_1 = (1 \cdots 0 \cdots \cdots \cdots \cdots)$
$c_j = (\cdots j \cdots \cdots \cdots \cdots \cdots)$
$c_t = (0 \cdots 1 \cdots \cdots \cdots \cdots)$

Fig. 7. The reverse basis vectors c_j.

The PBVM is constructed by the positive basis vectors, which is used to modify the data area of RS code. It can be expressed like Eq. (12).

$$\mathrm{PBVM} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix} = (\, I_k \quad | \quad P\,). \tag{12}$$

$$\mathrm{PBVM} \xrightarrow{\text{Row Transformation}} \begin{pmatrix} R & | & I_t \\ & \vdots & \\ 0 & \cdots & 0 \end{pmatrix}. \tag{13}$$

Where I_k is the k_{th} unit matrix, P with size $k * t$ is the corresponding matrix of parity area. The PBVM in matrix is shown in Fig. 8(a).

By performing elimination operation which is similar to the Gauss Jordan elimination on the PBVM, the parity matrix P is simplified to the simplest echelon matrix by row transformation, and the process is described in Eq. (13).

$$\mathrm{RBVM} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{pmatrix} = (\, I_t \quad | \quad R\,). \tag{14}$$

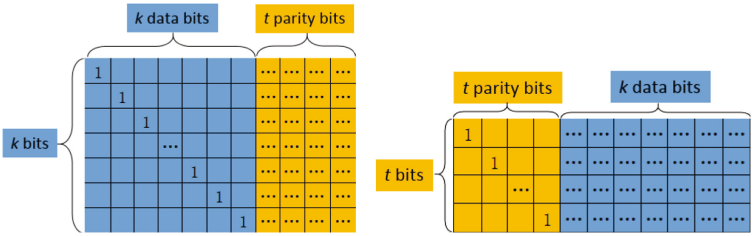
$$Q = \begin{cases} 0, T_i < T_0 \cap N_i = 0, \\ -1, T_i > T_0 \cap N_i = 0, \\ 1, T_i < T_0 \cap N_i = 1, \\ 0, T_i > T_0 \cap N_i = 1. \end{cases} \tag{15}$$

Thus the RBVM is obtained, which is used to modify the parity area of RS code. It can be expressed as follows in Eq. (14).

Where I_t is the t_{th} unit matrix, R with size $t * k$ is the corresponding matrix of data area. The RBVM is shown in Fig. 8(b).

The structure of RBVM is similar to PBVM, each row is a reverse basis vector c_j where $1 \leq j \leq t$. c_j indicates that the j_{th} bit is 1 in the parity bits and the rest bits are 0. The reverse basis vector c_j is represented in Fig. 7.

On the basis of the above works, the XOR operation can be performed on the encoded information stream with the generated basis vector matrixes, PBVM and RBVM. It is possible to control only one bit to be modified without changing the other bits. Once the data bit (or parity bit) changes, the corresponding parity bits (or data bits) remain updated simultaneously, so that the result is still a valid RS code without affecting the decoding rate.



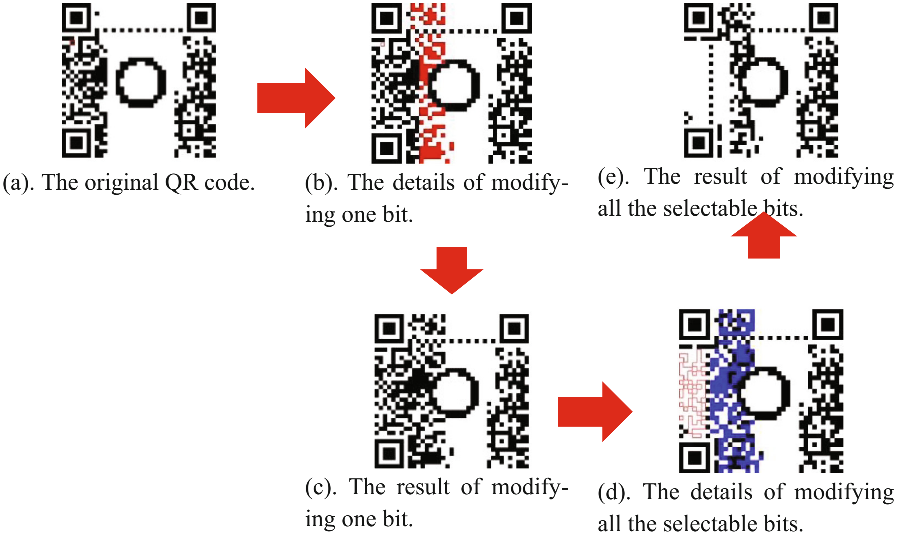
(a) PBVM (b) RBVM

Fig. 8. PBVM and RBVM

The proposed SPBVM. The proposed algorithm will select part of basis vectors in PBVM to construct the SPBVM, and the RBVM which generated by the SPBVM is more suitable for the fusion of background image.

In QR code, the length of data bits k is much larger than the length of parity bits t, so it is possible to select parts of basis vectors (b_{i1}, b_{i2}...b_{ij}...b_{il}, where $il > t$ and $1 \leq ij \leq k$) to participate in the elimination process. Therefore, a matrix called SPBVM is composed of the partial basis vectors selected from all basis vectors. According to the characteristics of human vision, the salient regions of background image are not expected to be modified. Therefore, the selection scheme of SPBVM combined with the salient region detection algorithm is proposed. First, the salient region detection is carried out. Then the corresponding basis vectors which is not in the salient region is selected to form SPBVM.

A simple example of the selection of basis vectors in SPBVM is shown in Fig. 9. (a) The original QR code is obtained, and the circle is the salient region. (b) Showing the details of the modification where the modified bit is represented by red box at the upper-left, and the affected code words are represented by red blocks. (c) The result of modifying one bit in parity area of (a). (d) Showing the details of the modification on the basis of (b). All the bits in selectable basis vectors are modified, and the modified bits are represented by red boxes, the affected code words are represented by blue blocks. (e) The result of modifying all the selectable bits. From the experimental results, we can see that the code words which affected by the modified bits all lay in the non-salient region.



(a). The original QR code.

(b). The details of modifying one bit.

(e). The result of modifying all the selectable bits.

(c). The result of modifying one bit.

(d). The details of modifying all the selectable bits.

Fig. 9. The application of SPBVM.

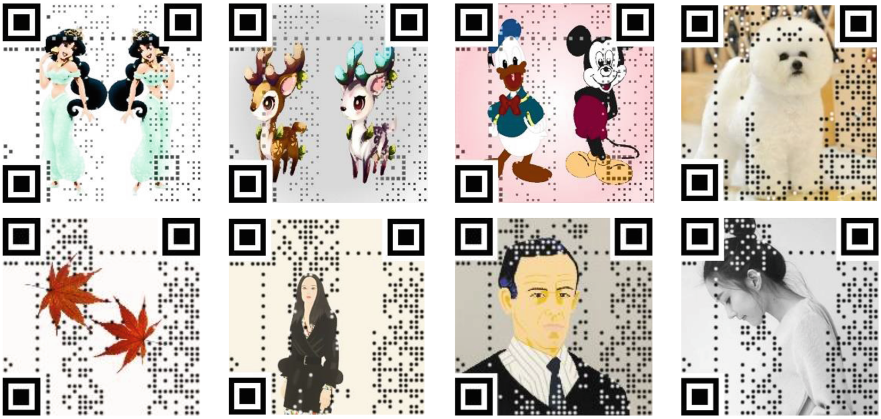
2.3 The Synthetic Strategy

To improve the visual quality after the above-mentioned operations, the synthetic strategy, which is proposed in [6], is used to achieve the fusion of background image and the QR code. The equation is described in (15). Where Q = 0 represents that the module is replaced completely by the background image. $Q = -1$ or 1 represents that the central area of the module is replaced by the corresponding area of the QR code, the other area of the module is replaced by the background image. And T_i represents the gray average value of the gray block, T_0 represents the binary threshold, N_i represents the module of QR code, $N_i = 1$ is the white module, and $N_i = 0$ is the black module.

3 Experimental Results

A dataset containing 100 images with different styles is ready for experiments. The message embedded into each QR code is “hello”. For all the aesthetic QR codes, the version number is 5 and error correction level is L, and some results generated by our method are shown in Table 2.

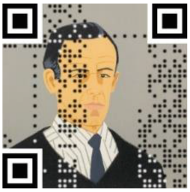
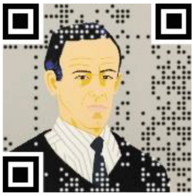
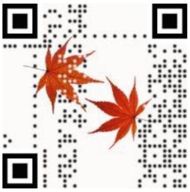
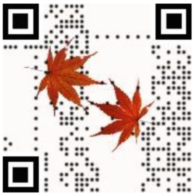
Table 2. The aesthetic QR code generated by our algorithm



3.1 Comparison Results with and Without Salient Region Detection

In this paper, an aesthetic QR code generation contrast experiment, with salient region detection and without salient region detection are designed. The experimental results are shown in Table 3.

Table 3. The generation of aesthetic QR codes with and without saliency region detection

Background image	Without salient region detection	With salient region detection
		
		

3.2 Correctness of QR Code Decoding

We evaluated the correctness of decoded messages on three different mobile devices (HUAWEI Honor 7, MI 5, iPhone 6) with three various QR code decoders (Wo Cha Cha, WeChat, Qrcode Scanner). The images in Table 2 which are generated by our algorithm are used to calculate the decoding rates. The results are reported in Table 4.

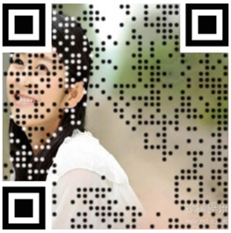
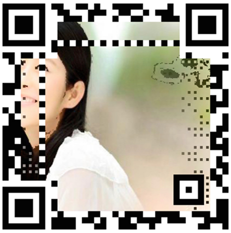
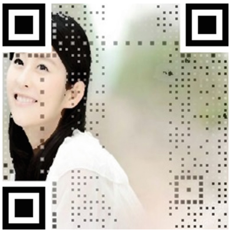
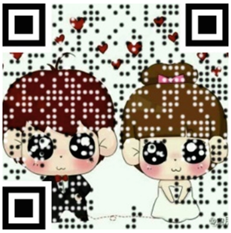
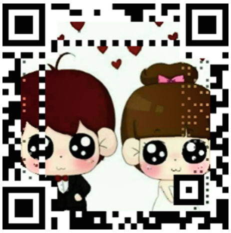
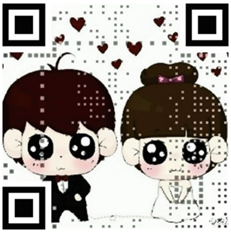
Table 4. Decoding rates on different mobile devices.

Mobile phone	APP	Decoding rates
HUAWEI Honor 7	Wo Cha Cha	100%
	WeChat	100%
	Qrcode Scanner	100%
MI 5	Wo Cha Cha	100%
	WeChat	92%
	Qrcode Scanner	100%
iPhone 6	Wo Cha Cha	100%
	WeChat	100%
	Qrcode Scanner	100%

3.3 Comparison of Different Aesthetic QR Code Algorithms

To evaluate the performance of our algorithm, we compare our aesthetic QR code algorithm to those in [6, 8]. The generated QR codes are presented in Table 5.

Table 5. The aesthetic QR codes generated by different algorithms.

[6]	[8]	Ours
		
		

4 Conclusion

In the algorithm, we make full use of the XOR characteristics of RS code, without sacrificing the error correction capacity of RS code. According to the acquired salient region, the basis vectors are selected from PBVM to form SPBVM. The associated RBVM can be conducted by carrying out the operation like Gauss Jordan elimination on SPBVM. The replacement in parity area with the RBVM generate the black and white modules which do not lie in the salient region. Experimental results show that the proposed aesthetic QR code algorithm can achieve a better visual effect.

Acknowledgments. This work was mainly supported by National Natural Science Foundation of China (No. 61370218).

References

1. Falcon, A.: 40 gorgeous QR code artworks that rock (2013). http://www.hongkiat.com/blog/qr-code-artworks/. Accessed 3 Mar 2017
2. Cox, R.: Qart codes (2012). http://research.swtch.com/qart. Accessed 20 Jan 2017
3. Cox, R.: Finite field arithmetic and Reed-Solomon coding (2012). http://research.swtch.com/field. Accessed 20 Jan 2017
4. Ono, S., Morinaga, K., Nakayama, S.: Two-dimensional barcode decoration based on real-coded genetic algorithm. In: Evolutionary Computation, pp. 1068–1073. IEEE (2008)
5. Baharav, Z., Kakarala, R.: Visually significant QR codes: image blending and statistical analysis. In: IEEE International Conference on Multimedia and Expo, pp. 1–6. IEEE (2013)
6. Li, L., Qiu, J., Lu, J., et al.: An aesthetic QR code solution based on error correction mechanism. J. Syst. Softw. **116**(C), 85–94 (2016)
7. Fujita, K., Kuribayashi, M., Morii, M.: Expansion of image displayable area in design QR code and its applications. In: Proceedings of the Forum on Information Technology Papers, vol. 10(4), pp. 517–520 (2011)
8. Lin, S.S., Hu, M.C., Lee, C.H., et al.: Efficient QR code beautification with high quality visual content. IEEE Trans. Multimedia **17**(9), 1515–1524 (2015)
9. Garateguy, G.J., Arce, G.R., Lau, D.L., et al.: QR images: optimized image embedding in QR codes. IEEE Trans. Image Process. **23**(7), 2842–2853 (2014)
10. Xu, X.Y.: Research on digital rights protection methods in digital publish, pp. 27–32. Hangzhou Dianzi University, Hangzhou (2016)
11. Cheng, M.M., Mitra, N.J., Huang, X., et al.: Global contrast based salient region detection. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 569–582 (2015)
12. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn., pp. 534–540. Publishing House of Electronics Industry, Beijing (2011). (in Chinese)
13. Wang, X., Georganas, N.D.: GLCM texture based fractal method for evaluating fabric surface roughness. In: Canadian Conference on Electrical and Computer Engineering, 2009, CCECE 2009, pp. 104–107. IEEE (2009)
14. Haralick, R.M., Shanmugam, K.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **3**(6), 610–621 (1973)

15. Kavitha, C., Rao, B.P., Govardhan, A.: Image retrieval based on color and texture features of the image sub-blocks. Int. J. Comput. Appl. **15**(7), 33–37 (2011)
16. Chen, Y.Q., Duan, J., Zhu, Y., Qian, X.F., Xiao, B.: Research on the image complexity based on texture features. Chin. Opt. **8**(3), 407–414 (2015). (in Chinese)
17. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)

Compact Cat Swarm Optimization Algorithm

Ming Zhao[1](✉), Jeng-Shyang Pan[2], and Shuo-Tsung Chen[3]

[1] School of Computer Science, Yangtze University, Jingzhou, China
hitmzhao@gmail.com
[2] School of Information Engineering,
Fuzhou University of Technology, Fuzhou, China
jengshyangpan@gmail.com
[3] Department of Mathematic, Tung Hai University, Taichung, Taiwan
shough32@hotmail.com

Abstract. A compact cat swarm optimization algorithm (cCSO) was proposed in this paper. it keeps the same search logic of cat swarm optimization (CSO), i.e. tracing mode and seeking mode, on the other hands, cCSO inherits the main feature of compact optimization algorithms, a normal probabilistic vector is used to generate new individuals, the mean and the standard deviation of the probabilistic model could lead cats to the searching direction in next step. Only a cat is adopted in the algorithm, thus, it could run with modest memory requirement. Experimental results show that cCSO has better performance than some compact optimization algorithms in some benchmark functions test. The convergence rate is also a highlight among compact optimization algorithms.

Keywords: Compact optimization · Cat swarm optimization
Normal probabilistic model · Memory saving · Differential factor

1 Introduction

Recently, compact optimization algorithms have got a rapid development. Georges R. Harik et al. proposed a novel compact genetic algorithm (cGA) [1] in 1999. It represented the population by using a probability distribution based on the set of solutions and runs with less memory than the simple GA. A real-valued Compact Genetic Algorithm (rCGA) [2] was presented for Embedded Microcontroller Optimization in 2008, instead of processing a real population, it first time employed a normal probability vector to generate a real-valued solution directly, and parameters of probability vector could be updated with specified rules, these updating rules could help genetic algorithm speed up the convergence rate and reduce large memory. Based on the same probability vector updating rules, Compact Differential evolutionary algorithm was designed in 2011 [3] by Ernesto Mininno et al. It uses the mutation and crossover typical of differential evolution to implement its search task. And this modest memory requirement makes the cDE algorithm have the ability to be embedded in some small computational power equipment [4]. After cDE, Ernesto Mininno et al. proposed the compact Particle Swarm Optimization algorithm [5] in 2013. After a solution (particle) Generated from the probability vector, a particle retains the search logic typical of Particle Swarm Optimization (PSO) algorithms [6] and begins to seek

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 33–43, 2018.
https://doi.org/10.1007/978-3-319-76451-1_4

for the best solution; it has good performance compared with other memory-saving algorithms. And it also suits for embedded micro-equipment.

As a new member of the bio-inspired swarm optimization algorithm, Cat Swarm Optimization algorithm (CSO) [7] was proposed by Chu et al., it showed a new cooperation searching logic which is tracing searching mode and seeking searching mode to solve continuous optimization problems. Tsai et al. developed CSO and presented a parallel cat swarm optimization (PCSO) algorithm [8] in 2008. Since then CSO were used widely to solve continuous optimization problems [9, 10].

CSO is a population-based optimization algorithm. And large memory usage is required. But in some special applications, such as micro-robot [11, 12], hardware and computation power are still limited. Based on these requirements, and inspired by cPSO, a compact cat swarm optimization algorithm (cCSO) was proposed in this paper. It also employs a probability model to generate new individuals; a novel differential factor was adopted in seeking mode. Experimental results show that cCSO algorithm has good performance and convergence rate compared with its population-based version and other compact optimization algorithms.

The remainder of this paper is organized as follows. Section 2 introduces sampling mechanism for Compact Optimization algorithms and cat swarm optimization. Section 3 derives the compact cat swarm optimization, Sect. 4 shows and analyzes the experimental results. Finally, conclusions are summarized in Sect. 5.

2 Related Work

In this section, the sampling mechanism of the compact algorithms with real-value encoding was presented in detail, and CSO was described in Sect. 2.2.

2.1 Virtual Population and Sampling Mechanism

As mentioned above, the main feature of the compact algorithm is population-less, A great deal of individuals would be generated in the whole evolutionary process. But in compact algorithms [1–3], the population is virtual. The virtual population is a probabilistic model of solutions statistic description instead of an actual population of solutions. Generally, normal probabilistic distribution model is employed to generate individuals. The model is named Perturbation Vector, and we indicated this vector with *PV*, The mean and the standard deviation are parameters of this model. Parameters are updated that it could get higher performance solution in the next step, it is encoded with a $n \times 2$ matrix in real-valued problems [8], and it is shown as formula (1):

$$PV^t = [\mu^t, \sigma^t] \tag{1}$$

Where μ and σ are respectively mean and standard deviation values of the corresponding Gaussian PDF of individuals. And the apex t is iteration generations.

In order to solve problems in different domains, without loss of generality, each variable should be normalized in the interval [–1, 1]. Thus, the Cumulative Distribution Function (CDF) for the corresponding PDF will be less than 1, there will be a error between the real value and the truncated value of each variable to the corresponding CDF value, then potential solutions which located out of the interval [–1,1] are needed to mapping to the interval [–1, 1], in order to solve this problem, an error function [13] was introduced, see formula 2.

$$\mathrm{erf(x)} = \frac{2}{\sqrt{\pi}}\int_{0}^{x} e^{-t^2} d_t \tag{2}$$

The PDF for truncated variable x could be presented as formula 3.

$$PDF(truncNorm(x_i)) = \frac{\sqrt{\frac{2}{\pi}}e^{-\frac{(x_i-u_i)^2}{2(\sigma_i)^2}}}{\sigma_i(erf(\frac{u_i+1}{\sqrt{2}\sigma_i}) - erf(\frac{u_i-1}{\sqrt{2}\sigma_i}))} \tag{3}$$

And the corresponding CDF is described by formula 4,

$$CDF(PDF(x)) = \int_{a}^{b} PDF(x)d_x \tag{4}$$

When a CDF value is generated by the CDF function, the corresponding variable x could be got by computing the inverse function of the CDF. And the computed variable x is required. The entire sampling mechanism is shown as Figs. 1 and 2.

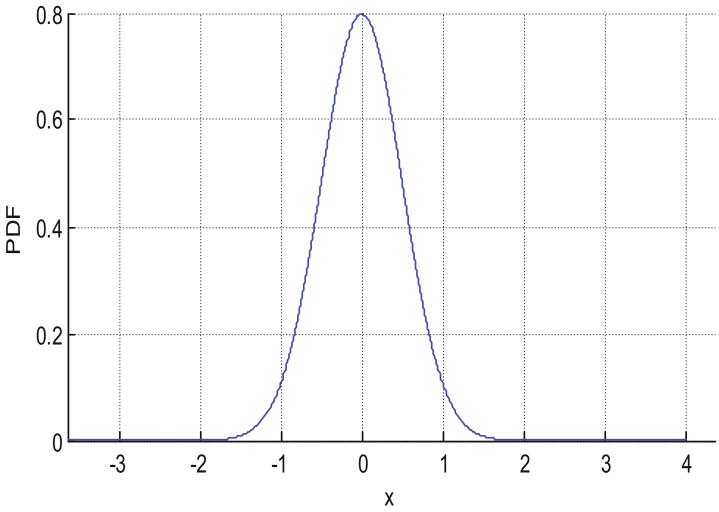


Fig. 1. The Gaussian probability distribution function curve

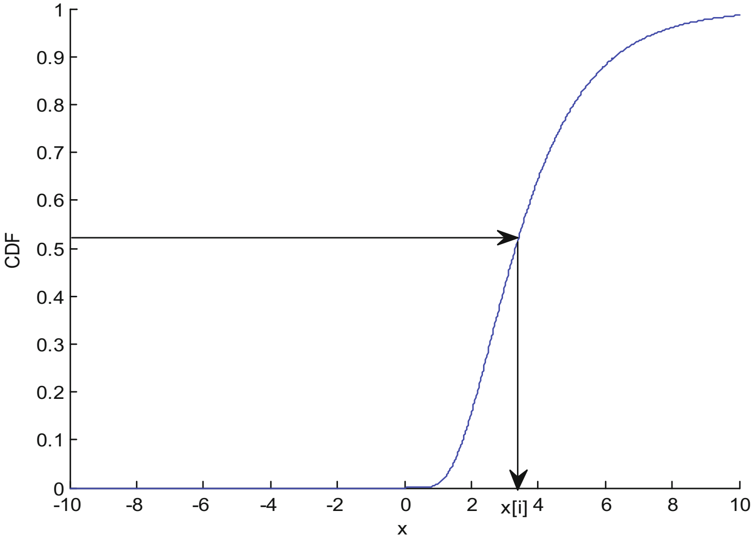


Fig. 2. The sampling mechanism

2.2 The Perturbation Vector Updating Rule

The Perturbation Vector of the virtual population is used to generate a new designed variable for the solution, we update the μ and σ to expect to generate better solution for the problem. And the updating rule for PV is given in formula (5) and (6).

$$u^{t+1}[i] = u^{t}[i] + \frac{1}{N_p}(winner[i] - loser[i]) \tag{5}$$

$$\sigma^{t+1}[i] = \sqrt{(\sigma^{t}[i])^2 + (u^{t}[i] - u^{t+1}[i])^2 + \frac{1}{N_p}(winner[i] - loser[i])} \tag{6}$$

In formula (5) and (6), where N_p is virtual population size. Normally, i is the dimension of the designed variable x. The details about formula (5) and (6) are interpreted in [2], *winner* and *loser* are two important vectors. It will be discussed in the Sect. 3.

2.3 Cat Swarm Optimization

Inspired by the behavior of cats capture their prey, Chu et al. presented the cat swarm optimization (CSO) algorithm [7] in 2007, in CSO, it employs cats to portray the solution sets. Each cat has two kinds of behaviors, which models into two modes, namely, seeking mode and tracing mode, a group of cats use cooperate to look for the best solution under a suitable proportion; it could get a good performance by this cooperation method. The details about CSO are presented in this section.

2.3.1 Seeking Mode

When cats in seeking mode, they will look around and only adjust their position slightly, and seek for the optimal opportunity to capture the prey. The algorithm used the following procedure to implement this function.

Firstly, each cat will duplicate its own position five times, these positions will be memorized in a seeking memory pool (SMP), then each value in SMP will be changed slightly by a mutagenic operator, a dimension of designed variable $x[i]$ could be selected to mutate, and the variation range can't be out of a fixed value, which was decided by SRM. And another parameter CDC will determine that how many dimensions of the solution will be mutated. The mutation operation will be presented as formula (7):

$$x[i] = x[i] + \Delta x[i] \tag{7}$$

The cat will select a point in SMP with best fitness to update its position.

2.3.2 Tracing Mode

When cats are in tracing mode, their behavior will be simulated to the particle in PSO algorithm, each cat traces the cat with the global best position to update its own velocity and position. However, cats approach the global best value quickly by learning from the group experience of the whole cat populations. And this update rule could speed up the convergence rate.

The updating rule of tracing mode could be presented as formula (8) and formula (9).

The combination of seeking mode and tracing mode could ensure that the cat swarm optimization algorithm convergence quickly and prevent the solution from local optimum.

$$\mathrm{V}_k(\mathrm{t}\ +1) = \omega \cdot \mathrm{V}_k(\mathrm{t}) + \mathrm{C} \cdot \mathrm{rand} \cdot [X_{gbest}(\mathrm{t}) - \mathrm{x}_k(\mathrm{t})] \tag{8}$$

$$X(t+1) = X(t) + V(t+1) \tag{9}$$

X_{gbest} is the best position of the cat with the best fitness; x_k is the position of cat_k. t is the iteration generations. And C is a constant and rand is a random number in the interval [0, 1].

Cats in seeking mode will select a position with the best fitness, and cats in tracing mode also will generate a cat with global best fitness, these two cats would be compared and the one with best fitness will be used to update the global best individual X_{gbest}. When iterations meet the termination condition, the final X_{gbest} is the solution for the problem.

3 Compact Cat Swarm Optimization Algorithm

A compact cat swarm optimization algorithm (cCSO) was proposed in this section, the same to existed compact algorithms, a perturbation vector PV with normal distribution probabilistic model is used. All designed variables x will be normalized to the intervals

[–1, 1]. Only a cat is adopted in cCSO, and it also has two modes, one is seeking mode and the other is tracing mode. The details would be presented in this section.

3.1 Initialization and Perturbation Vector Updating Rules

In initialization phase, the mean $\mu[i]$ of each dimension of designed variable are assigned 0, and standard deviation $\sigma[i]$ are assigned 10. Means $\mu[i]$ standard deviations $\sigma[i]$ are updated according to formula (5) and (6). There are two very important vectors *winner* and *loser*. Obviously, μ and σ are updated by a factor $\frac{1}{N_p}(winner - loser)$, the vector $(winner - loser)$ is indicator for local best solution for the next iteration. It could be seen in Fig. 3.

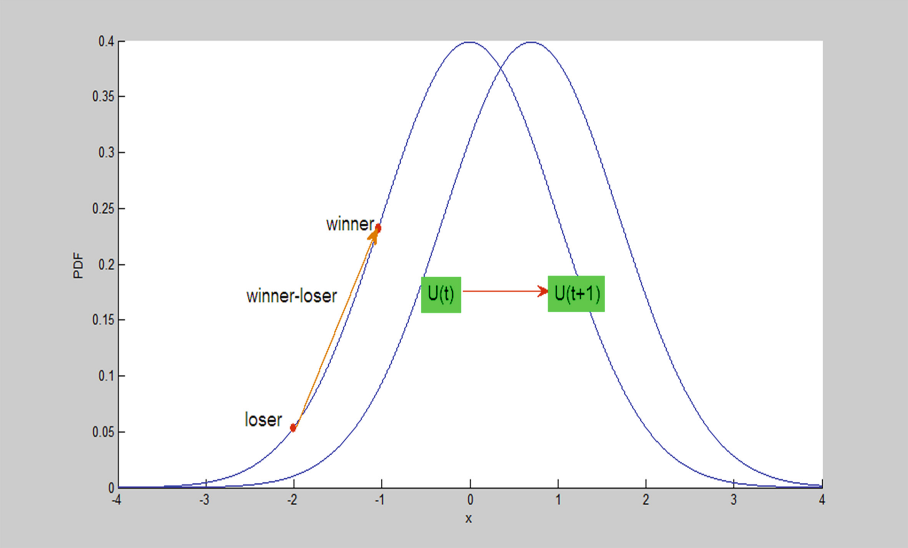


Fig. 3. The interpretation for updating rule with *winner* and *loser*

3.2 Seeking Mode

When the cat is in seeking mode, it only updates the position of cats, the formula (9) will be implemented in CSO algorithm, in cCSO, the updating rule is different from the formula (7) and was presented as formula (10).

$$x = x + \frac{1}{N_p}(winner - loser) \tag{10}$$

In formula (12), the vector $winner - loser$ could face all directions, maybe it has the larger probability to approach the best position quickly.

The number rate of cats in seeking mode and in tracing mode is set to 49:1 [8]. But in proposed cCSO, only a cat is in seeking mode, in order to mimic the search logic of the corresponding CSO algorithm, the updating rule will be implemented 49 times.

3.3 Tracing Mode

In tracing mode, the cat employed need to updating its position and velocity. And the updating rule is similar to Particle Swarm Optimization algorithm [9]. The difference between cCSO and PSO lies that cat only traces the global best position to update its own velocity and position. The updating rules are seen in formula (8) and (9).

3.4 The Implementation Procedure and Details

In the proposed cCSO, the Perturbation Vector also be initialized firstly, see [10], and a rand will determine the cat will go into seeking mode or seeking mode.

When the cat is in seeking mode, cat's position will be changed by differential factor (*winner-loser*), and cat's position also is updated by *winner*, and, μ and σ would be updated by, μ and σ each run. The final *winner* will be left to x_{gb}, While the cat is in tracing mode, it will be updated similar to cPSO. For the sake of clarity, the pseudo code for cCSO is shown as Fig. 4.

4 Experimental Results and Analyze

In order to test the performance of the proposed cCSO algorithm, five test benchmark numerical functions [11, 12] were employed in this paper. The set of test functions contains the testbed from IEEE CEC 2008 [17]. It is shown as follows.

Shifted sphere function:

$$f_1(x) = \sum_{i=1}^{D} z_i^2 \; z_i = x - o; D = [-100, 100] \tag{11}$$

Schwefel's Problem:

$$f_2(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2 z_i = x - o; D = [-100, 100] \tag{12}$$

Shifted Ackley's function:

$$f_3(x) = -20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2*pi*z_i)} + 20 + e, \; z_i = x - o; D = [-32, 32] \tag{13}$$

```
  t=0;
  initialization u and sigma;
  random generate $x_{gb}$ , cat.x and cat.v
  p=rand;
  if (p>0.2) cat. Mode=seeking else cat. Mode=tracing
  while (t<maxiteration)
  if (cat. Mode=seeking)
  {generate xlb from PV
  cat.x(t+1) =cat.x(t)+c1*rand*(cat.x(t)-xlb)
  [winner, loser] =compete(cat.x(t+1), xlb);
  Updating u and sigma according to formula 1.5 and 1.6
  Cat.x=winner;
  [winner,loser]=compete(cat.x,xgb);
  xgb=winner;}
    Else {generate xlb from PV
      Cat.v(t+1) =w* cat.v(t)+c2*r2*(xgb-cat.x(t)); Cat.x=cat.x+cat.v
  [winner, loser] =compete(cat.x(t+1), xlb);
      Updating u and sigma according to formula 1.5 and 1.6
      [winner, loser] =compete(cat.x(t+1), xgb);
      xgb=winner;}
  t=t+1;
end while
```

Fig. 4. The algorithm flow chart for cCSO

Shifted Griewank's function:

$$f_4(x) = \sum_{i=1}^{n} \frac{z_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \; z_i = x - o; \quad D = [-600, 600] \tag{14}$$

Shifted Rastrigin's function:

$$f_5(x) = 10n + \sum_{i=1}^{n} [z_i^2 - 10\cos(2\pi z_i)]z_i = x - o, D = [-5, 5] \tag{15}$$

Considerate that cCSO is a development version of cPSO, we compared with PSO, cPSO. And reference to paper [4, 17], parameters of compared algorithms are listed in Table 1.

Table 1. Parameters list for all compared algorithms in this study

Algorithm	Parameter	Original paper	Algorithm	Parameter	Original paper
CSO	w = 0.9–0.4 N_p= 40 c_1 = c_2= 2.0	[7]	PSO	$\phi_1 = -0.2$ $\phi_2 - 0.07$ $\phi_3 = 3.74$ $\gamma_1 = \gamma_2 = 1$ $N_p = 60$	[18]
cPSO	$\phi_1 = -0.2$ $\phi_2 - 0.07$ $\phi_3 = 3.74$ $\gamma_1 = \gamma_2 = 1$ $N_p = 300$	[4]	cCSO	$w = -0.2$ $c_2 = 2.0$ $c_1 = 3.74$ $N_p = 300$	

Experiments were implemented in MATLAB on a personal computer with a Pentium(R) Dual-core E6600 CPU, 3.06GHZ, 2.96 GB RAM. The demo system is set at windows XP platform.

In order to get fair comparison results, for each compared algorithm, all benchmark functions were computed over 30 times.

This section is organized as follows: firstly, a summary for memory usage is listed as Table 2. Then, comparison among different algorithms is presented. and finally, results analysis for cCSO are summarized.

Table 2. Memory employments for all compared algorithms

Algorithm	Components	Memory slots
cCSO	Compact CSO based structure 1 sampling	5
cPSO	Compact PSO based structure 1 sampling	5
PSO	PSO structure	$2N_P$
CSO	CSO based structure	$2N_P$

4.1 Memory Usage for All Compared Algorithms

The proposed cCSO algorithm inherits the main feature of compact optimization algorithms, it has modest memory requirement. Because it has the same *PV* structure, the CSO also has the similar data structure to CSO, so the total memory usage is need to store five solutions. The detail about each compared algorithm is listed as Table 2.

4.2 Comparisons for Convergence Result

Results in Table 3 shows that experimental results for cCSO and compared compact algorithms. From Table 3, the proposed cCSO displayed a pretty good performance than cPSO.

Table 3. Comparison for convergence results

Test problem	CSO	PSO	cPSO	cCSO
f1	0.000e+00 ± 0.00e+00	1.252e+04 ± 5.30e+03	6.471e+01 ± 2.28+01	**3.261e+01 ± 1.13e+01**
f2	0.000e+00 ± 0.00e+00	4.231e+04 ± 6.97e+03	2.560e+03 ± 2.36e+03	*1.869e+03 ± 1.306e+02*
f4	8.880e–17 ± 1.13e–19	1.638e+01 ± 1.21e+00	3.728e+00 ± 3.71e–01	6.573e+00 ± 2.0356e–01
f5	0.000e+00 ± 0.00e+00	0.000e+00 ± 0.00e+00	9.636e–08 ± 3.07e–08	**1.000e–08 ±** 1.74–09
f6	0.000e+00 ± 0.00e+00	2.886e+02 ± 3.27e+01	2.940e+01 ± 7.94e+00	5.40e–7 ± 1.935e–08

Based on the same *PV* to generate new individual, and the combination for two searching logic method can help it get the more effectively solution.

5 Conclusion

This paper proposes a novel optimization algorithm, namely cCSO, this algorithm employs the search logic of CSO with a virtual population, a probabilistic representation of the population is used to generate new individual and guide the search direction for the next iteration. the algorithm could run with modest requirement and it also could ensure that the searching of cats becomes more effectively. The experimental results showed that it played a better performance cPSO. It fits to solve problems which happen in environment with limited hardware.

References

1. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. IEEE Trans. Evol. Comput. **3**(4), 287–297 (1999)
2. Mininno, E., Cupertino, F., Naso, D.: Real-valued compact genetic algorithms for embedded microcontroller optimization. IEEE Trans. Evol. Comput. **12**(2), 203–219 (2008)
3. Mininno, E., Neri, F., Cupertino, F., Naso, D.: Compact differential evolution. IEEE Trans. Evol. Comput. **15**(1), 32–54 (2011)

4. Iacca, G., Neri, F., Mininno, E.: Opposition-based learning in compact differential evolution. In: Evo Applications 2011 Part I, Lecture Notes in Computer Science, vol. 6624, pp. 264–273. Springer (2011)
5. Neri, F., Mininno, E., Iacca, G.: Compact particle swarm optimization. Inf. Sci. **239**, 96–121 (2013)
6. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
7. Chu, S.C., Tsai, P.W., Pan, J.S.: Cat swarm optimization. In: Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence, pp. 854–858 (2006)
8. Tsai, P.-W., Pan, J.-S., Chen, S.-M., Liao, B.-Y.: Enhanced parallel cat swarm optimization based on the Taguchi method. Expert Syst. Appl. **39**(7), 6309–6319 (2012)
9. Pradhan, P.M., Panda, G.: Solving multi objective problems using cat swarm optimization. Expert Syst. Appl. **39**(3), 2956–2964 (2012)
10. Wang, Z.-H., Chang, C.-C., Li, M.-C.: Optimizing least-significant-bit substitution using cat swarm. Inf. Sci. **192**(1), 98–108 (2012)
11. Jung, M.-J., Myung, H., Lee, H.-K., Bang, S.: Ambiguity resolving in structured light 2D range finder for SLAM operation for home robot applications. In: Proceedings of the IEEE Workshop on Advanced Robotics and its Social Impacts, pp. 18–23 (2005)
12. Okazaki, A., Senoo, T., Imae, J., Kobayashi, T., Zhai, G.: Real-time optimization for cleaner-robot with multi-joint arm. In: Proceedings of the International Conference on Networking, Sensing and Control, pp. 885–890 (2009)
13. Gautschi, W.: Error function and fresnel integrals, In: Abramowitz, M., Stegun, I.A. (eds.) Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, pp. 297–309 (1972)
14. Cody, W.J.: Rational Chebyshev approximations for the error function. Math. Comput. **23** (107), 631–637 (1969)
15. Neri, F., Cotta, C., Moscato, P.: Handbook of Memetic Algorithms, Studies in Computational Intelligence, vol. 379. Springer, Berlin Heidelberg (2011)
16. van den Bergh, F., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 225–239 (2004)
17. Tang, K., Yao, X., Suganthan, P.N., MacNish, C., Chen, Y.P., Chen, C.M., Yang, Z.: Benchmark functions for the CEC'2008 special session and competition on large scale global optimization. Technical report
18. Pedersen, M.E.H.: Good parameters for particle swarm optimization. Technical report HL1001, Hvass Lab. (2010)

Cryptanalysis and Detection Systems

Copy-Move Forgery Detection Based on Local Gabor Wavelets Patterns

Chao-Lung Chou[1(✉)] and Jen-Chun Lee[2]

[1] Department of Computer Science and Information Engineering,
Chung Cheng Institute of Technology, National Defense University, Taoyuan, Taiwan
chaolung.chou@gmail.com

[2] Department of Electrical Engineering, Chinese Naval Academy, Kaohsiung, Taiwan
i923002@gmail.com

Abstract. Nowadays digital images are more and more easily to be modified or tampered intentionally by most people due to the rapid development of powerful image processing software. Various methods of digital image forgery exist, such as image splicing, copy-move forgery, and image retouching. Copy-move is one of the typical image forgery methods, in which a part of an image is duplicated and used to replace another part of the same image at a different location. In this paper, we proposed a block-based passive detect copy-move forgery detection method based on local Gabor wavelets patterns (LGWP) with the advantages of high performance texture analysis of Gabor filter and rotation-invariant ability of uniform local binary pattern (LBP). Experiment results demonstrate the ability of the proposed method to detect copy-move forgery and precisely locate the duplicated regions, even when the forgery images are distorted by JPEG compression, blurring, brightness adjustment and rotation.

Keywords: Copy-move forgery · Image forgery detection
Local Gabor wavelets patterns (LGWP)

1 Introduction

Due to the rapid development of powerful image processing software, digital images are more and more easily to be modified or tampered intentionally by most people. Copy-move is one of the typical image forgery methods, in which a part of an image is duplicated and used to replace another part of the same image at a different location.

Image forgery detection is to detect whether if an image is affected by some kind of manipulations such as copy or move, image splicing and image touching. Image forgery detection techniques can be broadly categorized into active and passive approaches. The active approaches embedded additional information in an image in advance and then extracted that to discriminate its integrity. The most common methods are digital watermarks and digital signatures. The passive approach, on the other hand, is capable of detecting image manipulation without priori information. Therefore, the passive approaches are more practical in real-life applications.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 47–56, 2018.
https://doi.org/10.1007/978-3-319-76451-1_5

A common image forgery detection consists of four stages [1]. The first stage is typically pre-processing, in which usually including color conversion and overlap or non-overlap image partition. This stage is used to reduce the computation complexity and increase processing efficiency. The second stage is feature extraction which is to select representative image features for further discrimination. The third stage is to match extracted features in the image and determine if it is manipulated. The matching stage is either by block-based or keypoint-based. Finally, the results of tempered region will be localized and displayed.

Copy-move forgery detection techniques can be categorized into block-based and keypoint-based approaches. The block-based approach splits an image into either overlap or non-overlap blocks. Then, the features are extracted from these blocks and compared the similarity between blocks within the image. Generally, the feature extraction techniques for block-based are in the form of frequency transform, texture and intensity, and etc. Fridrich et al. [2] proposed the first block matching detection scheme based on the discrete cosine transform (DCT). Popescu and Farid [3] proposed a copy-move forgery detection method by using principal component analysis (PCA) instead of DCT. Hsu and Wang [4], Lee [5] using Gabor wavelet features to extract image block pattern information. Davarzani [6] et al. using multiresolution local binary pattern (MLBP) to extract image block pattern information. These two pattern information are known for their robustness to geometric distortions and illumination variations.

On the other hand, keypoint-based methods extract distinctive local features from entire image. Each feature is presented with a set of descriptor produced within a region around the features. Both features and descriptors in the image are classified and matched to each other to find the forgery regions. The most popular keypoint-based approaches are scale invariant feature transform (SIFT) [7, 8] and speed up robust features (SURF) [9].

In this paper, we propose a passive copy-move forgery detection method based on local Gabor wavelets patterns (LGWP). The image is converted into a gray-scale image and divided into overlapping fixed-size blocks. The proposed LGWP descriptor is applied to each block for local features extraction. The lexicographical sorting algorithm is adopted to reduce matching time while comparing image blocks features. Finally, regions of image forgery is detected through the identification of similar block pairs.

The remainder of the paper is organized as follows. In Sect. 2, the LGWP descriptor is introduced, and Sect. 3 describes the proposed method. Section 4 present the results of experiments and evaluate the performance of the proposed method. The conclusions are presented in Sect. 5.

2 Local Gabor Wavelets Patterns

Gabor filters are well-known to be particularly appropriate for texture analysis due to its similarity to those of the human visual system (HVS). Daugman [10] proposed the 2D Gabor functions by a series local spatial bandpass filters to accurate 2D space and 2D spatial frequency location. It is found that the 2D Gabor filter provide robustness against image brightness and contrast varying and now are being used extensively in image processing applications such as iris recognition and fingerprint recognition.

The general form of a 2D Gabor filter is expressed as follows:

$$G_{\sigma,f,\theta}(x,y) = g_{\sigma}(x,y)\exp[2\pi\, if(x\,cos\theta + y\,sin\theta)] \tag{1}$$

where

$$g_{\sigma}(x,y) = \frac{1}{2\pi\sigma^2}\exp[-\frac{(\mathrm{x}^2+\mathrm{y}^2)}{2\sigma^2}] \tag{2}$$

and $J = \sqrt{-1}$, $g_{\sigma}(x,y)$ is the Gaussian function with scale parameter σ, f is the frequency parameter, θ is the orientation parameter. Let $I(x,y)$ denotes a grayscale image and $G_{\sigma,f,\theta}(x,y)$ represent a Gabor filter. The Gabor magnitude output of an image $I(x,y)$ is obtained by convolution of each block with the Gabor filter until the entire image is traversed. The magnitude responses $M_{\sigma,f,\theta}(x,y)$ of the Gabor filter can be computed as follow:

$$M_{\sigma,f,\theta}(x,y) = \sqrt{C_R^2(x,y)_{\sigma,f,\theta} + C_I^2(x,y)_{\sigma,f,\theta}} \tag{3}$$

where $C_R^2(x,y)_{\sigma,f,\theta}$ and $C_I^2(x,y)_{\sigma,f,\theta}$ denote the real and imaginary components of the discrete convolutions results of I(x, y) and $G_{\sigma,f,\theta}(x,y)$.

The local object appearance and shape can be characterized using the local magnitude directions distribution. We define $\theta_k = \frac{\pi(k-1)}{n}, k = 1, \ldots, n$ as the orientation k in total n orientations. In most cases, one would use 2D Gabor filters with eight different orientations. That is $n = 8$ and $\theta_{k=1\ldots8} = \left\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \ldots, \frac{7\pi}{8}\right\}$. Suppose there are total N sub-blocks in $I(x,y)$, the average Gabor filter respond magnitude of all directions in the same frequency and scale can be calculated as $M_{\bar{\theta}}(x,y) = \frac{1}{N}\sum M_{\theta_k}(x,y)$. The orientation that corresponds to the strongest textural information point $d(x,y)$ is defined as follows

$$d(x,y) = \arg\max_{k=1,\ldots,n}\left\{M_{\theta_k}(x,y)\right\} \tag{4}$$

The local Gabor wavelets patterns $LGWP(x,y)$ defined as follows:

$$LGWP(x,y) = \begin{cases} 1, & if\left(M_{\theta_k}(x,y) - M_{\bar{\theta}}(x,y)\right)2^{mod(n-d,k)} \geq 0 \\ 0, & otherwise \end{cases} \tag{5}$$

The modular operation in (5) is used to keep rotation-invariant textural information.

Suppose an image sub-block with $k = 8$ and $M_{\bar{\theta}}(x,y) = 100$. Figure 1(a) shows the Gabor filter respond magnitude of all 8 directions and the maxima magnitude is 167. Figure 1(b) shows the LGWP code (10001010) using (5). Notice that all points of that

Gabor filter respond magnitude greater than 100 are coded with 1. Figure 1(c) shows the image sub-block rotated 90° and Fig. 1(d) shows the corresponding LGWP code (10001010). It is obvious the LGWP code is efficient to locate the rotation and is robust to resist such attack.

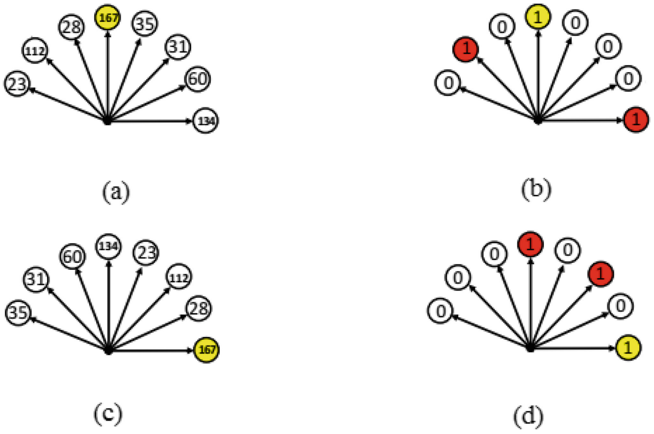


Fig. 1. Examples of LGWP code. (a) Original Gabor filter respond magnitude, (b) The LGWP code of (a), (c) Gabor filter respond magnitude after rotated 90°, (d) The LGWP code of (c).

3 The Proposed Method

In the proposed method, original image first divided into overlapping blocks of a fixed size, then the similarity of these blocks are detected, and finally displayed the possible duplicated regions. A flow-chart of the proposed forgery detection method is shown in Fig. 2.

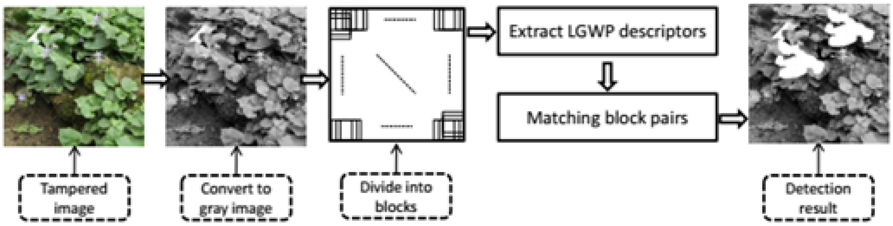


Fig. 2. The flow-chart of the proposed algorithm.

3.1 Image Pre-processing

First, the color image is converted into the gray scale image I. Then the $M \times N$ grayscale image I is divided into overlapping sub-blocks. Each block is denoted as B_{ij},

$$B_{\mathrm{ij}}(x, y) = I(x + j, y + i) \tag{6}$$

where $x, y \in \{0, \cdots, B - 1\}, i \in \{1, \cdots, M - B + 1\}$, and $j \in \{1, \cdots, N - B + 1\}$. Hence, the grayscale image I is divided into $(M - B + 1) \times (N - B + 1)$ overlapping blocks.

3.2 Feature Extraction with LGWP

In this paper, we consider 8 directions for each block. In that, total $2^8 = 256$ features can be used to represent each block. To reduce computation complexity, we using so called uniform patterns proposed by Ojala et al. [11] to extracted features from circular blocks after LGWP features extraction. A local binary pattern (LBP) is called uniform if its uniformity measure is at most 2. The 36 unique rotation invariant binary patterns that can occur in the circularly symmetric 8 neighbors as shown in Fig. 3. By applying uniform local binary pattern, the LGWP feature vector can efficient reduced from 256 to 36 and maintain rotation-invariant ability at the same time.

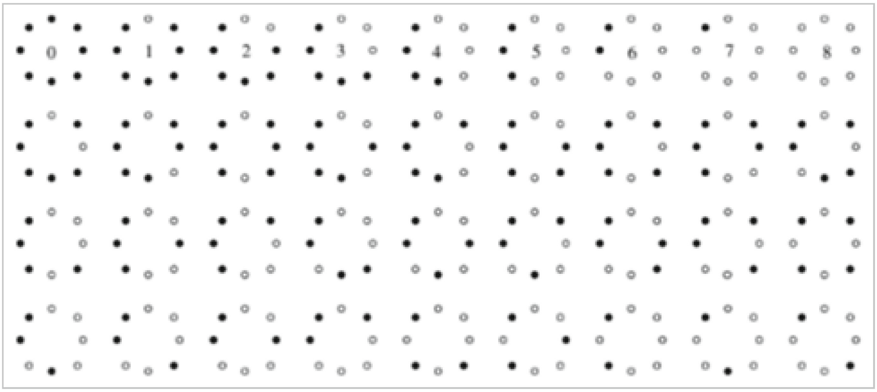


Fig. 3. The 36 unique rotation invariant binary patterns that can occur in the circularly symmetric 8 neighbors [11].

3.3 Matching Block Pairs

The matching techniques enhances the computational complexity during the search of identical values in a large size image. For block-based image forgery detection, sorting, hash, correlation and Euclidean distance are most common approaches [1]. In this paper, we use the lexicographical sorting technique to detect potentially tampered region through the adjacent identical pairs of blocks. The similar feature vectors are stored in neighboring rows after lexicographical sorting, such that the features of duplicated block pairs appear successively. The blocks were compared using Euclidean distance as follows:

$$B_{\mathrm{distance}}\left(\widehat{V}_i, \widehat{V}_{i+j}\right) = \sqrt{\left(x_i - x_{i+j}\right)^2 + \left(y_i - y_{i+j}\right)^2} \tag{7}$$

where (x, y) is the center of the corresponding block and $\widehat{V}_i$, $\widehat{V}_{i+j}$ are sorted adjacent feature vectors derivative from original feature vector $V_i = (f_1, f_2, \ldots, f_{36})$.

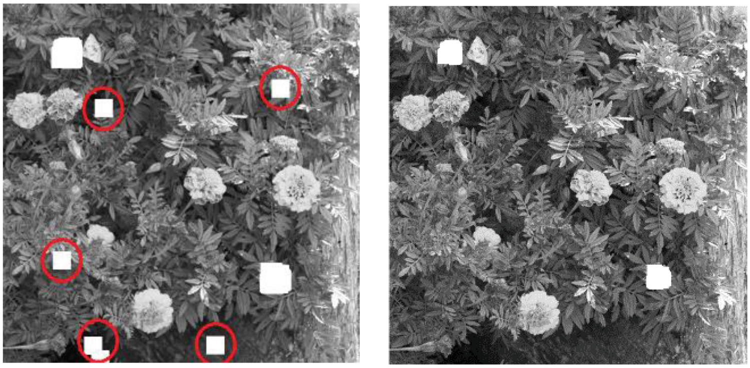
The more similar between blocks, the smaller value of $B_{\text{distance}}\left(\widehat{V}_i, \widehat{V}_{i+j}\right)$ is calculated. Hence, a predefine threshold T_s is given to indicate their similarity. We define (i) is the smallest distance between the ith and the nearby features in vector $\widehat{V}_i$ lower than T_s as follows:

$$D(i, \sigma) = min\{D(i;i - j), \ldots, D(i;i - 1), D(i;i + 1), \ldots, (i;i + j)\} \tag{8}$$

In addition, there is high possibility that the similarity of nearby blocks feature vectors is very close. Thus, we compared only blocks in which the position distance from other blocks exceeds distance threshold T_d.

3.4 Post-processing

Generally, all detected blocks, including the original and forged blocks, are marked into white (pixel value = 255) to generate the detection result. Figure 4(a) shows an example of the early detection results with some distortion (marked in red circle). To obtain accurate forgery regions, all blocks are further calculated their pairwise alike based on the area of these blocks using 4-connected components labeling method. The difference Fig. 4(b) shows the final detection results after post-processing from Fig. 4(a).



(a) (b)

Fig. 4. Examples of detection results, (a) early results with distortions, (b) final results after post-processing.

4 Experimental Results

In the experiments, the proposed method is evaluated using publicly available CoMoFoD database [12]. The database consists of 260 forgery images with two different sizes 512 × 512 and 3,000 × 2,000. Here, we use the 512 × 512 size for all experiments. Images are grouped into 5 groups of manipulation: translation, rotation, scaling, combination and distortion. Different types of post-processing methods, such as JPEG compression, blurring, noise adding, color reduction etc., are applied to all forged and original images.

All experiments were performed on a personal computer with a 3.2 GHz CPU, 4 GB memory, with MATLAB 8.5 environment. To illustrate the performance of the proposed algorithm, we referenced correct detection ratio (CDR) indicates the performance of the algorithm in terms of accurately locating the pixels of copy-move regions in the tampered image defined as follows:

$$\mathrm{CDR} = \frac{\textit{The detected tampered region}}{\textit{The tampered region}} \tag{9}$$

At first, we test the detection performance without post-processing. Figure 5 shows the example of detection results.

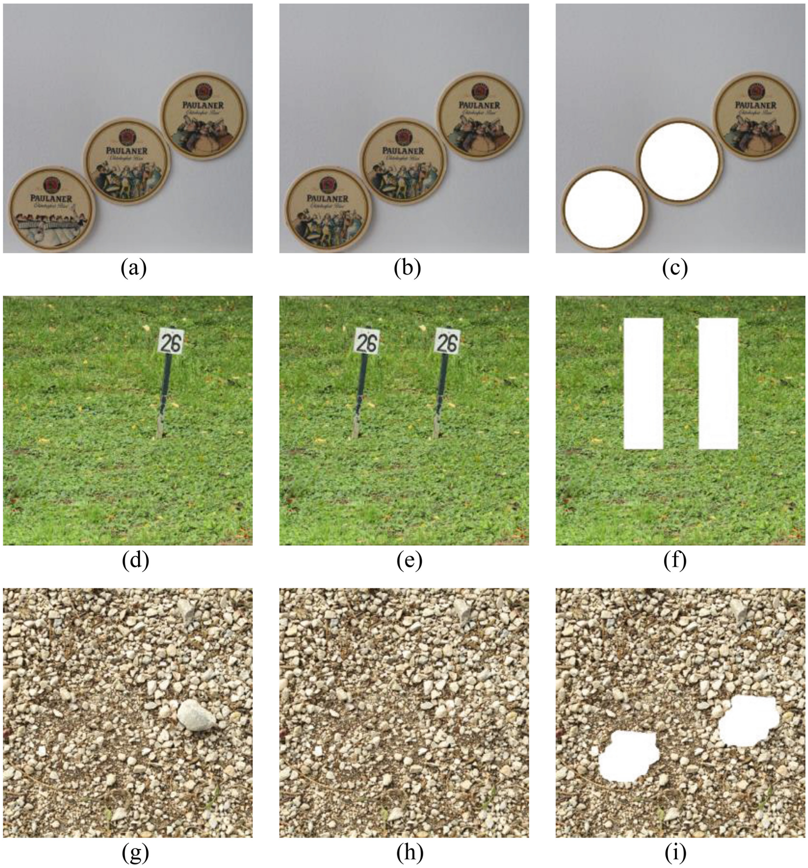
The statistical detection rates without post-processing for sub-blocks of various sizes of 16 × 16, 32 × 32, and 48 × 48 are presented in Table 1. The proposed method performs well in blocks sizes of 16 × 16 than other size because some portions of the forged regions are so small that they cannot be detected when using larger block sizes. Thus, we use the block size at 16 × 16 for further experiments.

The ability to resist post-processing attacks is fundamental to copy-move forgery detection methods. The most common post-processing attacks are JPEG compression, brightness adjustment, blurring and rotation. To evaluate the robustness and effectiveness of the proposed method in resisting above post-processing attacks, the experimental results were compared with [5] in JPEG compression (quality factor = 20, 30, 50, 70, 90), brightness adjustment ([0.01, 0.95], [0.01, 0.90] and [0.01, 0.8]), Gaussian blurring ($\sigma^2 = 0.005$ and 0.0005) and rotation (0°, 2°, 20°, 45°, 60°, 90°, 150° and 180°) showing in Table 2.

As shown in Table 2, the proposed algorithm achieved high correct detection ratios for JPEG compression with a quality factor above 70 and also provides excellent robustness against changes in image brightness, as evidenced by the reliable detection performance achieved in the [0.01, 0.8] range.

The forged images blurring using Gaussian blurring with standard deviation equals 0.005 and 0.0005. Both detection results are in high performance.

The case in which the forged images regions is copied, rotated and moved to another position in the same image without distorting it using any other techniques. The duplicated regions are rotated by angles selected with 0°, 2°, 20°, 45°, 60°, 90°, 150° and 180°. Figure 6 shows the examples of image with different rotating angles. As Table 2 shows, the proposed method is robust against rotation attack at different rotating angles and outperformed [5].



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 5. Detection results without post-processing (a), (d), (g) original image, (b), (e), (h) forgery image, and (c), (f), (i) detection results.

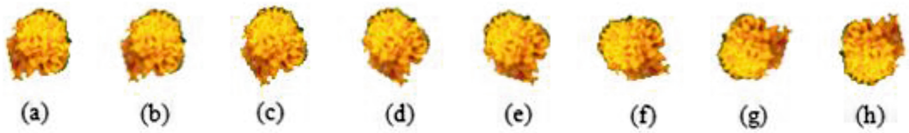


Fig. 6. Examples of rotation attacks for a copy-move forgery region (a) original (b) 2° (c) 20° (d) 45° (e) 60° (f) 90° (g) 150° (h) 180°.

Table 1. Copy-move forgery detection results of the proposed method without post-processing.

Block size	CDR
16 × 16	0.991
32 × 32	0.974
48 × 48	0.967

Table 2. Comparison of detection results of forged images by JPEG compression, brightness adjustment, gaussian blurring and rotation.

Post-processing attacks		The proposed method	[5]
JPEG compression (Quality factor)	90	0.975	0.970
	70	0.910	0.920
	50	0.862	0.840
	30	0.570	0.520
	20	0.350	0.320
Brightness adjustment	[0.01, 0.95]	0.990	0.986
	[0.01, 0.90]	0.990	0.975
	[0.01, 0.80]	0.990	0.953
Gaussian Blurring (σ^2)	0.005	0980	0.976
	0.0005	0.958	0.946
Rotation angles	0°	0.991	0.988
	2°	0.942	0.93
	20°	0.830	0.12
	45°	0.910	N/A
	60°	0.810	0.09
	90°	0.991	N/A
	150°	0.840	N/A
	180°	0.991	0.38

5 Conclusions

Image forgery detection is a rapidly growing research area, especially on passive techniques. Block-based approach is more popular approach due to its suitability with various feature extraction techniques and the capability to achieve a high matching performance. In this paper, we proposed a passive block-based image copy-move forgery detection method based on local Gabor wavelets patterns (LGWP) with the advantages of high performance texture analysis of Gabor filter and rotation-invariant ability of uniform local binary pattern (LBP). Experiment results demonstrate the efficacy and robustness of the proposed algorithm in detecting copy-move forgery, while forgery images is under JPEG compression, brightness adjustment, blurring, and rotation.

References

1. Warif, N.B.A., Wahab, A.W.A., Idris, M.Y.I., Ramli, R., Salleh, R., Shamshirband, S., Choo, K.-K.R.: Copy-move forgery detection: survey, challenges and future directions. J. Netw. Comput. Appl. **75**, 259–278 (2016)
2. Fridrich, J., Soukal, D., Lukas, J.: Detection of copy–move forgery in digital images. In: Proceedings of Digital Forensic Research Workshop, pp. 19–23 (2003)
3. Popescu A., Farid, H.: Exposing digital forgeries by detecting duplicated image regions. Technical report TR2004-515, Department of Computer Science, Dartmouth College (2004)
4. Hsu, H.C., Wang, M.S.: Detection of copy-move forgery image using Gabor descriptor. In: Proceedings of International Conference on Anti-Counterfeiting, Security and Identification (ASID), pp. 1–4 (2012)
5. Lee, J.-C.: Copy-move image forgery detection based on Gabor magnitude. J. Vis. Commun. Image Represent. **31**, 320–334 (2015)
6. Davarzani, R., Yaghmaie, K., Mozaffari, S., Tapak, M.: Copy-move forgery detection using multiresolution local binary patterns. Forensic Sci. Int. **231**, 61–72 (2013)
7. Amerini, I., Ballan, L., Caldelli, R., Bimbo, A.D., Serra, G.: A SIFT based forensic method for copy-move attack detection and transformation recovery. IEEE Trans. Inf. Forensics Secur. **6**(3), 1099–1110 (2011)
8. Christlein, V., Riess, C., Jordan, J., Riess, C., Angelopoulou, E.: An evaluation of popular copy-move forgery detection approaches. IEEE Trans. Inf. Forensics Secur. **7**(6), 1841–1854 (2012)
9. Bo, X., Junwen, W., Guangjie, L., Yuewei, D.: Image copy-move forgery detection based on SURF. In: Proceedings of International Conference on Multimedia Information Networking and Security, pp. 889–892 (2010)
10. Daugman, J.: Two-dimensional analysis of cortical receptive field profiles. Vision. Res. **20**, 846–856 (1980)
11. Ojala, T., Pietikainen, M., Maèenpaèa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
12. CoMoFoD database Homepage. http://www.vcl.fer.hr/comofod. Accessed 23 Oct 2017

A Hybrid Intrusion Detection System for Contemporary Network Intrusion Dataset

Jheng-Mo Liao, Jui-Sheng Liu, and Sheng-De Wang(✉)

Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan
sdwang@ntu.edu.tw

Abstract. We propose a hybrid intrusion detection approach to detect network anomalies. The proposed approach uses a feature discrete method and a cluster analysis algorithm to separate the training samples into two groups, normal and anomaly groups, and then a new classification model is built to improve the performance of the sub group classification. We discretize the features of training samples by the method considering the interdependence between features and labels. Class information is added into the attributes to enhance the clustering results. For the anomaly group, several representative features are selected to construct a classification model to improve the overall classification performance. Two efficient machine learning algorithms, the Decision Tree algorithm and the Bayesian Network algorithm, are adopted in our experiment. The experiment results show that our method can increase both the normal and anomaly detection rate, precision and accuracy. For the classification of new types of modern attacks, our approach also can improve the overall accuracy.

Keywords: Intrusion detection system · Machine learning
Contemporary attack detection

1 Introduction

Network intrusion detection has been widely studied for past decades. In various types of intrusion detection datasets, we concentrate on network flow based dataset analysis. Mostly attributes in the network traffic are continuous values. The discretization techniques transforms the continuous value into categorical value. This transformation provides a different perspective for us to analyze the dataset. Garcia et al. [1] introduce various discretization methods and show that some of them can improve the classification accuracy. From the overall point of view to consider the intrusion detection system. Garcia Teodoro et al. [2] present a survey of various techniques for anomaly based intrusion detection. Buczak and Guven [3] provide a detail overview of cyber security intrusion detection approaches and compare the pros and cons of various techniques for both anomaly and misuse based intrusion detection issues.

1.1 Cluster Analysis

Clustering is a widely used anomaly detection method to divide the different behaviors of groups [6, 7, 11–14]. Guo et al. [4] propose a two-stage architecture for anomaly

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 57–69, 2018.
https://doi.org/10.1007/978-3-319-76451-1_6

detection. The first stage separates the data into normal and abnormal behavior by using K-Means. The second stage uses KNN as the main machine learning algorithm. The abnormal behavior is classified by an anomaly element and the normal behavior is detected by a misuse element. This hybrid approach can effectively detect anomalies. Om et al. [8] design a hybrid system combining K-Means and two classifiers, the KNN algorithm and the Naïve Bayes algorithm, to reduce the false alarm rate in anomaly detection. Al-Yaseen et al. [9] propose a multi-level hybrid intrusion detection system for misuse detection. They use modified K-Means to obtain multiple clusters to construct different SVM and ELM models for each class. Anita et al. [10] propos a hybrid framework combining K-Means, KNN and Decision Table to improve the detection results.

1.2 Feature Selection

There are two major categories of method for feature selection, i.e., wrapper based and filter based. Wrapper based feature selections search the best performance feature subset for a particular learning classifier. This method usually spends extremely high computing time. Filter based feature selections evaluate feature subsets by computing the attribute information, i.e., information gain, distance and correlation. Kaur et al. [16] compare the performance of nine different feature selection methods for the NSL-KDD dataset with the Naïve Bayes classifier. Desale and Ade [18] propose an approach to intersect three different feature selection algorithms to discuss classification results, i.e., correlation based feature selection with genetic search, information gain with ranker and correlation attribute evaluator with ranker. Haq et al. [17] propose an ensemble framework consisting of three feature selection algorithms and three machine learning algorithms. They choose three different feature subsets for the three machine learning algorithms respectively and vote the detection results to improve the final detection performance. Pervez and Farid [19] propose a wrapper based like feature selection algorithm for SVM classifier and the NSL-KDD dataset.

2 Methodology

We propose a new approach for intrusion detection systems to detect modern attacks. The concept is using an enhanced machine learning model to improve the performance of classification. As Fig. 1 shows, our approach can be roughly divided into the preprocessing phase, the feature selection phase and the constructing classification model phase.

2.1 Dataset

The UNSW-NB15 dataset [20] was created by using IXIA tool to extract contemporary network activities contain both normal and new type of attacks in the Cyber Range lab of the Australian Centre for Cyber Security (ACCS). The whole dataset has 2,540,044 records, which are directly captured from raw network traffics. It has 49 features with two types of classes, attack categories and binary labels. The features involve five

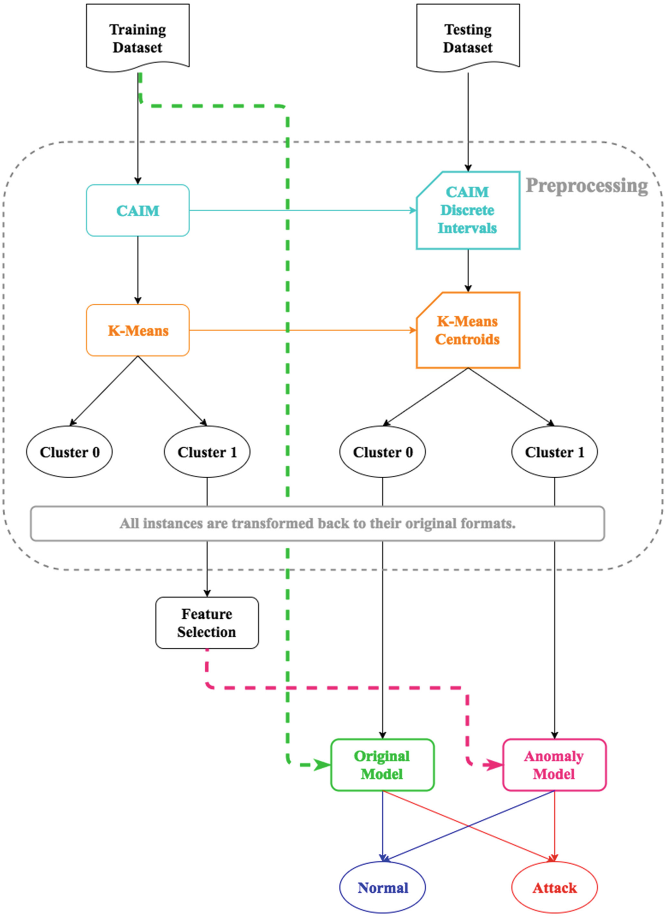


Fig. 1. The flow chart of the proposed approach

categories, namely flow features, basic features, content features, time features and additional generated features. In recent network environment, nine up-to-date attack activities are simulated, i.e., Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode and Worms. Moreover, a training dataset and a testing dataset are divided from the whole dataset by the collectors. The training dataset has 175,341 records and the testing dataset has 82,332 records. There is no duplicate instance in these two datasets. 42 features are selected from the original features.

Compared to the previous intrusion detection dataset, KDD99, the UNSW-NB15 dataset has some important differences as follows. First, in UNSW-NB15, the training dataset and the testing dataset have the same distribution and are statistically similar to each other. Second, most of the features in UNSW-NB15 are different from KDD99, even though they are extracted from the network flow. Last, UNSW-NB15 is more difficult to analyze than KDD99 because normal traffics and contemporary abnormal traffics have extremely similar behavior. The difficulty of classification in UNSW-NB15 was presented [20, 22]. On average, each machine learning algorithm on the overall accuracy is reduced by about 10%. Many past intrusion detection methods may not work in the contemporary network environment.

2.2 Discretization

Discretization is a kind of procedure for transforming a continuous attribute into a nominal attribute. It can be divided into unsupervised types and supervised types. Unsupervised discretization methods discretize attributes without using the information of the class labels. Equal width (EW) and equal frequency (EF) are commonly used unsupervised discretization algorithms. Supervised methods discretize attributes by using the interdependence between attributes and class labels, i.e., Information Entropy Maximization (IEM) and Maximum Entropy.

Class attribute interdependence maximization discretization (CAIM) algorithm [5] is one of the supervised discretization methods. CAIM has two main considerations to determine the discretization scheme. First, the discretization scheme should have a minimum number of the discrete intervals. The other one is that the information loss caused by the discretization process should be reduced to minimum. The value of CAIM criterion is defined as:

$$CAIM(C, D|F) = \frac{\sum_{r=1}^{n} \frac{max_r^2}{M_{+r}}}{n} \tag{1}$$

where C is the class variable in the sample, D is a discrete variable which we choose in this round, F is the attribute under consideration, n is the number of discrete intervals, r is an iteration variable of discrete intervals from 1 to n, max_r is the maximum value in the r-th discrete interval among all classes and M_{+r} is the total number of values belonging to the r-th discrete interval.

The higher the value of CAIM criterion represents a better way to discretize the attribute. The brief steps of CAIM are introduced as follows. In the beginning, the maximum value and the minimum value of each attribute in the training dataset are recorded. For each attribute, the inner boundary of the attribute is tentatively added as the division point and the value of CAIM criterion is calculated. Each division boundary is considered greedy until the largest value of CAIM criterion is found. We determine the division boundary with the largest value of CAIM criterion as the best discrete scheme.

In our experiment, binary labels, Normal and Attack are assigned to be discrete class variables. We use the upper bound of discrete results as new value in each attribute. An example of an instance before and after discretization is shown in Table 1. The new instance values are still continuous types. However, this new values contain the class information.

Table 1. Samples of raw training data and results of converting the data

Original instance
0.121478,6,4,258,172,74.08749,252,254,14158.94238,8495.365234,0,0,24.2956,8.375,30.177547, 11.830604,255,621772692,2202533631,255,0,0,0,43,43,0,0,1,0,1,1,1,1,0,0,0,1,1,0
New instance
59.9999890.121478,12.0,16.0,1468.0,2454.0,17241.37888,255.0,254.0,18843182.0,49025.00781,3.0,3.0, 84371.496,56716.824,1460480.016,289388.2697,255.0,4294958913.0,4294881924.0,255.0,0.000245, 0.0002035,3.45e-05,57.0,75.0,1.0,3915.0,15.0,0.5,10.0,6.0,2.0,9.0,0.5,0.5,0.5, 11.0,15.0,0.5

2.3 Cluster Analysis

The main goal of cluster analysis is to group instances together based on the information of attribute values. Instances within the same group are similar but not similar to instances of other groups. K-Means is a centroid based approach to separate the sample into K disjoint groups. The brief steps of K-Means are as follows. At first, K initial temporary centroids are randomly given. The distances between instances and temporary centroids are iteratively computed. Each instance selects the nearest temporary centroid to group into K clusters. Then, new means for each cluster are recalculated to replace the temporary centroids. Repeat assigning each instance to the group with the closet cluster and recalculating the centroids until all centroids are not changed anymore. There are many different measures for distance calculation. We choose the most widely used measure, Euclidean distance. The formula for the Euclidean distance is:

$$Distance(p,\ q) \ = \ \sqrt{\sum\nolimits_{i=1}^{n} (q_i - p_i)^2} \tag{2}$$

where p and q represent two meanings, one is two points in the Euclidean space and the other one is two instances in the sample.

Continue with the previous step, the new format of the sample contains the class information. Using this advanced information, two groups can be separated more dissimilarly to each other. The results of clustering the training dataset with or without applying CAIM are different. In the following sections, we take these two cases as K-Means combined with CAIM and K-Means respectively.

2.4 Correlation Based Feature Selection

Because discrete values miss some information, it can cause poor classification results. The values of each attribute are retransformed to the original values after clustering. Then, we examine the sub samples belonging to the anomaly cluster. Some irrelevant features that may lead to erroneous decisions are picked out. Only choosing important features to improve the classification performance. We use correlation based method with genetic search to select features.

Correlation based Feature Selection [15], CFS is a heuristic method that ranks the importance of each feature. An excellent feature is defined by two criteria. First one is that the feature is highly correlated with the class labels. The other one is that the feature should not be correlated with other features. In CFS, the formula for the feature subset evaluation function is defined as:

$$Merit = \frac{k * r_{cf}}{\sqrt{k + k * (k - 1) * r_{ff}}}$$

where *Merit* is the score of a feature subset, r_{cf} is an average relevant score between features and classes and r_{ff} is an average relevant score between features and features.

After assigning scores to features, we have to search the most efficient feature subset. There are a lot of search methods can choose to reduce feature subset, i.e., BestFirst, Greedy and Random. We select a powerful algorithm, genetic search in our experiment. Genetic algorithms are widely used to solve global optimization problems in various domains. Each possible feature subset is regarded as a chromosome and go through four stages, namely the initialization stage, the selection stage, the crossover stage and the mutation stage. In accordance with the natural selection process to obtain the best solution. In our approach, we assign the maximum generated variable to 20 and the crossover probability to 0.6. In K-Means combined with CAIM, four features are selected, i.e., sttl, swin, smean and is_sm_ips_ports. These four features are important to recognize attacks in the anomaly cluster. The number of features for constructing machine learning model is reduced from 42 to 4. We can observe that four features are related to the source configuration. On the other hand, sttl and swin are found when we using K-Means without applying CAIM.

2.5 Machine Learning Algorithm

Two different classification algorithms are adopted to construct decision modules. We choose two different based methods, entropy based algorithms and probability based algorithms.

(1) *Decision Tree*

Decision Tree is a tree like structure. It uses attribute values to create rules and expresses decisions. The leaves represent classifications and the branches represent the attribute conditions cause the decisions. Each node is determined by computing the information entropy of each feature. ID3 and C4.5 are the two widely used algorithms to automatically construct Decision Tree models. We choose C4.5 because it is modified from ID3 and it is improved several ID3 weak points. In C4.5, the gain ratio is used to make decisions. The gain ratio is normalized the information gain. The attribute with the highest gain ratio is selected to be branching criterion. We use J48 to construct our Decision Tree model. J48 is a C4.5 implemented by Java in WEKA. In the previous phase, four features are selected to establish the classification mechanism.

(2) *Bayesian Network*

Bayesian Network is a probabilistic graphical model. It considers the attributes and their conditional dependencies. The concept of Bayesian Network is establishing a directed acyclic graph. In the graph, the child node is identified to be dependent on their parents. Compared with the Naïve Bayes algorithm, the Bayesian Network algorithm usually has better prediction. Because Naïve Bayes has a strong assumption, it assumes that each condition is independent. This assumption is too unrealistic on many issues. The drawback of Bayesian Network is that it is hard to determine the relationships between attributes. It may be decided by expert knowledge or using complicated algorithms. In our experiment, we used the simple estimator and the hill climber search algorithm to construct our Bayesian Network model.

3 Implementation and Results

The whole experiments were conducted on an Ubuntu 14.04 LTS with Intel Core i7-3770 K running at 3.5 GHz and 32 GB RAM. The implementation was coded using the python language and machine learning algorithms of WEKA tool was applied. PyCAIM is an open source code in python language. This project completely implement the CAIM supervised discretization. We import PyCAIM into our code to discretize the training dataset. WEKA is also an open source project, it contains numerous machine learning algorithms for data mining. The algorithms can be used directly by the tool or be imported from the Java code. We chose J48 to implement the Decision Tree algorithm and BayesNet to implement the Bayesian Network algorithm. The WEKA version used in our project is 3.8.0.

3.1 Evaluation Metrics

We use the following four metrics: (1) Detection rate: synonymous with true positive rate, the ratio of attack correctly detected; (2) False alarm rate: synonymous with false positive rate, the ratio of normal incorrectly detected; (3) Precision: the number of instances detected as attack are correctly detected; (4) Accuracy: the number of normal and attack instances that are correctly detected divided by the total number of samples.

3.2 Results with Cluster Analysis

Clustering is applied to separate the data into two groups. Figure 2 present the clustering results. Cluster1 is regarded as the anomaly cluster. The result of K-Means combined with CAIM reduces 5.5% of samples in the anomaly cluster. Most of them are normal instances and they are classified into Cluster0. We can notice that the testing dataset has similar clustering effects by applying CAIM as the training dataset.

3.3 Results with Enhanced Anomaly Model

According to clustering results, there are a total of 36351 instances in Cluster1. They will be classified for binary classes (normal and attack) by our enhanced anomaly model. In this section, we evaluate the performance of the enhanced anomaly model. The same sub dataset is detected by the original model and the enhanced anomaly model. The original model is built by all features and the entire training dataset. As Table 2 show, in this sub dataset, the performance of the enhanced anomaly model is better than the original model. The original Decision Tree algorithm has 99.2%

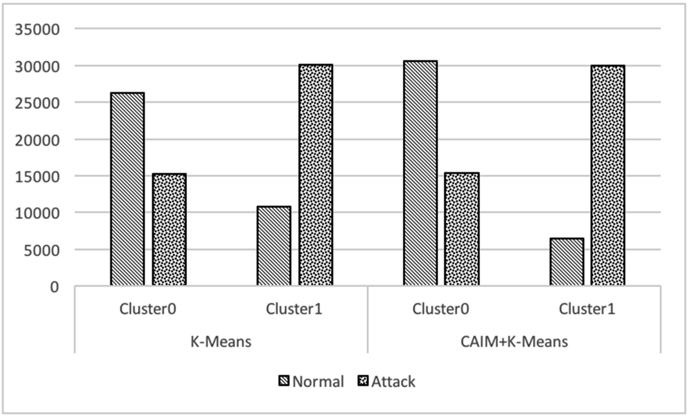


Fig. 2. Results of clustering the training dataset with or without applying CAIM

detection rate with 45.5% false alarm rate. In our enhanced anomaly model constructed by the Decision Tree classifier, the detection rate can be increased to 99.6% with 43.1% false alarm rate. On the other hand, the original Bayesian Network algorithm has 99.8% detection rate but the false alarm rate up to 80.9%. However, in our enhanced anomaly model constructed by the Bayesian Network classifier, the detection rate is 99.9% with only 53% false alarm rate. The enhanced anomaly model is increased the detection rate while reduced the false alarm rate. The results simultaneously indicate that the enhanced anomaly model has higher accuracy than the original model. In summary, using the enhanced anomaly model is better than using the original model if the new coming instance is belonging to the anomaly cluster.

Table 2. Comparison of anomaly detection performance for the anomaly cluster

	Decision tree		Bayesian network	
	Original model	*Proposed method*	*Original model*	*Proposed method*
Detection rate	99.2	**99.6**	99.8	**99.9**
False alarm rate	45.5	**43.1**	80.9	**53**
Precision	91	**91.5**	85.2	**89.8**
Accuracy	91.32	**92.09**	85.54	**90.51**

3.4 Results with Proposed Method

In order to verify the performance of our hybrid approach, three different conditions are considered. First, the entire testing dataset is classified by original machine learning algorithms without any modified. Second, we only apply the K-Means centroids to clustering the testing dataset into two clusters. Then, the newly created anomaly model is constructed by using the sub training dataset in the cluster1 with selected feature subset. The cluster0 sub testing dataset is detected by the original machine learning model and the cluster1 sub testing dataset is classified by the newly created anomaly model. This condition will be called basic K-Means in the following sections. The basic K-Means case evaluate the effect of the CAIM. The last condition is our method. The CAIM is applied before clustering.

As shown in Table 3, when we use the Decision Tree classifier, the basic K-Means has better detection rate than the original model. However, the high false alarm rate causes the poor overall accuracy. Our method solves the problem of high false alarm rate. The cluster1 we choose to construct the enhanced anomaly model can behave better than the basic K-Means. The accuracy of our method is 87.36% and the false positive rate is 25.95%. Even though the original Decision Tree algorithm has outstanding performance in detecting anomalies, we still improve it by using our enhanced anomaly model.

Table 3. Comparison of anomaly detection performance for the overall system in the decision tree algorithm

	Original decision tree	Basic K-Means	Proposed method
Detection rate	97.96	98.35	**98.24**
False alarm rate	26.36	28	**25.95**
Precision	81.99	81.14	**82.26**
Accuracy	87.03	86.5	**87.36**

In the Bayesian Network algorithm, the overall performance is almost the same whether or not CAIM is applied. CAIM only effects a little. However, the hybrid clustering framework can significantly improve overall performance, as shown in Table 4. In short, our method has 85.45% accuracy at 26.73% false alarm rate. We can detect anomalies more precise than the original Bayesian Network algorithm. In these three conditions, our accuracy is the highest.

Table 4. Comparison of anomaly detection performance for the overall system in the Bayesian network algorithm

	Original Bayesian network	Basic K-Means	Proposed method
Detection rate	95.36	95.41	**95.39**
False alarm rate	31.58	27	**26.73**
Precision	78.72	81.2	**81.39**
Accuracy	83.25	85.31	**85.45**

3.5 Additional Testing

As before, we use K-Means combined with CAIM to obtain two groups. After we obtain the anomaly cluster, we still use the CFS feature selection with genetic search to find representative features to construct the enhanced anomaly model. Eight features are selected in our approach, i.e., proto, service, dpkts, sbytes, sload, smean, ct_state_ttl and ct_src_dport_ltm.

At first, we use Decision Tree as the classifier to evaluate our approach. The overall detection rates, false alarm rates and accuracies are described in Table 5. The detection rates for ten categories are shown in Table 6. For the detection rates of each attack category, most of them are improved by our method. On the whole, our approach has the highest overall detection rate and accuracy with the lowest false alarm rate. We improve 2.35% detection rate and decrease 1.21% false alarm rate.

Table 5. Comparison of misuse detection performance for the overall system in the decision tree algorithm

	Original decision tree	Proposed method
Detection rate	75.34	**77.69**
False alarm rate	25.49	**24.28**
Accuracy	74.96	**76.84**

Table 6. Comparison of detection rates for each category in the decision tree algorithm

	Original decision tree	Proposed method
Normal	74.51	**75.79**
Generic	96.66	**97.57**
Exploits	83.28	**91.61**
Fuzzers	48.99	**49.21**
DoS	11.67	**11.45**
Reconnaissance	80.15	**81.01**
Analysis	0	**0**
Backdoor	15.44	**6.17**
Shellcode	72.75	**69.31**
Worms	59.09	**63.64**

Next, the Bayesian Network algorithm is regarded as the machine learning classifier. Applying the enhanced anomaly model can increase 4.22% detection rate and decrease 5.46% false alarm rate. The overall accuracy is improved 4.77% as shown in Table 7. The detection rates of each attack still can not all be improved as shown in Table 8.

3.6 Compared with Other Work

Moustafa and Slay [21] use the same dataset to train a model. A new feature selection method by using association rule mining combined with central points are proposed for

Table 7. Comparison of misuse detection performance for the overall system in the Bayesian network algorithm

	Original Bayesian network	Proposed method
Detection rate	71.69	**75.91**
False alarm rate	46.83	**41.37**
Accuracy	63.37	**68.14**

Table 8. Comparison of detection rate for each category in the Bayesian network algorithm

	Original Bayesian network	Proposed method
Normal	53.17	**58.63**
Generic	96.18	**96.19**
Exploits	52.68	**43.12**
Fuzzers	59.82	**74.67**
DoS	43.82	**15.38**
Reconnaissance	74.51	**74.97**
Analysis	0.3	**2.07**
Backdoor	25.9	**0.17**
Shellcode	71.69	**77.51**
Worms	86.36	**86.36**

the intrusion detection system. The central points can reduce the processing time when we select the most frequent feature subset for anomaly detection. The selected features can improve the evaluation of decision models. Eleven features are chosen, i.e., state, dttl, synack, swin, dwin, ct_state_ttl, ct_src_ltm, ct_srv_dst, sttl, ct_dst_sport_ltm and djit.

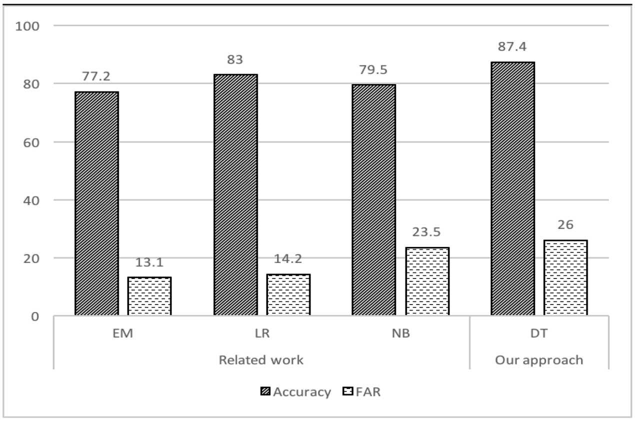


Fig. 3. Performance comparison of related methods

EM clustering (EM), Logistic Regression (LR) and Naïve Bayes (NB) are applied to evaluate the performance. The overall accuracy and the false alarm rate are supposed to be their evaluation criteria. As Fig. 3 shows, our method has the best overall accuracy among all detection models.

4 Conclusions

An enhanced anomaly model has been proposed to improve the performance of the suspicious group. In order to pick out this suspicious part of samples, we apply cluster analysis combined with CAIM discretization algorithm to group instances of the high probability of being abnormal. Some important features are selected from this part of samples to construct an anomaly model. On the other hand, the remaining samples are classified by the original model. The original model is constructed with the entire dataset and all features. We use the Decision Tree algorithm and the Bayesian Network algorithm to learn the classification mechanism to evaluate our approach. These two machine learning algorithms always have overall excellent detection results. The experiment results show that our approach can precisely detect abnormal activities in both machine learning models for the anomaly detection. In our approach, the instances belonging to the anomaly cluster can be accurately classified by the enhanced anomaly model.

References

1. García, S., Luengo, J., Sáez, J.A., López, V., Herrera, F.: A survey of discretization techniques: taxonomy and empirical analysis in supervised learning. IEEE Trans. Knowl. Data Eng. **25**, 734–750 (2013)
2. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., Vázquez, E.: Anomaly-based network intrusion detection: techniques, systems and challenges. Comput. Secur. **28**, 18–28 (2009)
3. Buczak, A.L., Guven, E.: A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutorials **18**, 1153–1176 (2015)
4. Guo, C., Ping, Y., Liu, N., Luo, S.-S.: A two level hybrid approach for intrusion detection. Neurocomputing **214**, 391–400 (2016)
5. Kurgan, L.A., Cios, K.J.: CAIM discretization algorithm. IEEE Trans. Knowl. Data Eng. **16**, 145–153 (2004)
6. Lin, W.-C., Ke, S.-W., Tsai, C.-F.: CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl. Based Syst. **78**, 13–21 (2015)
7. Yin, C., Zhang, S., Wang. J., Kim, J.-U.: An improved K-means using in anomaly detection. In: Computational Intelligence Theory, Systems and Applications (CCITSA) (2015)
8. Om, H., Kundu, A.: A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. In: Recent Advances in Information Technology (RAIT) (2012)
9. Al-Yaseen, W.L., Othman, Z.A., Zakree, M., Nazri, A.: Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. In: Expert Systems with Applications, vol. 67, pp. 296–303, January 2017
10. Chordia Anita, S., Gupta, S.: An effective model for anomaly IDS to improve the efficiency. In: Green Computing and Internet of Things (ICGCIoT) (2015)

11. Aissa, N.B., Guerroumi, M.: A genetic clustering technique for anomaly based intrusion detection systems. In: Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) (2015)
12. Liu, L., Wan, P., Wang, Y., Liu, S.: Clustering and hybrid genetic algorithm based intrusion detection strategy. Indonesian J. Electr. Eng. **12** (2014). TELKOMNIKA
13. Eslamnezhad, M., Varjani, A.Y.: Intrusion detection based on MinMax K-means clustering. In: Telecommunications (IST) (2014)
14. Varuna, S., Natesan, P.: An integration of K-Means clustering and Naïve Bayes classifier for intrusion detection. In: Signal Processing, Communication and Networking (ICSCN) (2015)
15. Hall, M.A.: Correlation-based Feature Selection for Machine Learning, Ph.D. dissertation, University of Waikato, New Zealand, April 1999
16. Kaur, R., Kumar, G., Kumar, K.: A comparative study of feature selection techniques for intrusion detection. In: Computing for Sustainable Global Development (INDIACom) (2015)
17. Haq, N.F., Onik, A.R., Shah, F.M.: An ensemble framework of anomaly detection using Hybridized Feature Selection Approach (HFSA). In: SAI Intelligent Systems Conference (IntelliSys) (2015)
18. Desale, K.S., Ade, R.: Genetic algorithm based feature selection approach for effective intrusion detection system. In: Computer Communication and Informatics (ICCCI) (2015)
19. Pervez, M.S., Farid, D.M.: Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In: Software, Knowledge, Information Management and Applications (SKIMA) (2014)
20. Moustafa, N., Slay, J.: The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Secur. J. Glob. Perspect. **25**, 18–31 (2016)
21. Moustafa, N., Slay, J.: A hybrid feature selection for network intrusion detection systems: central points and association rules. In: Australian Information Warfare Conference, December 2015
22. Moustafa, N., Slay, J.: The significant feature of the UNSW-NB15 and the KDD99 datasets for network intrusion detection systems. In: Proceedings of the 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS 2015), November 2015
23. Fawcett, T.: An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006)

Mitigating DoS Attacks in SDN Using Offloading Path Strategies

Tai-Siang Huang, Po-Yang Hsiung, and Bo-Chao Cheng(✉)

Department of Communications Engineering, National Chung-Cheng University,
Chia-Yi, Taiwan
x030060@gmail.com, paul100dtj@gmail.com, bcheng@ccu.edu.tw

Abstract. Software-Defined Networks (SDNs) were created to facilitate the management and control of the network. However, the security problem is still unresolved. To avoid the DoS attacks caused by links exceeding the bandwidth load (such as traffic flooding and security loopholes), the most simple mitigation solution is to offload the data by transferring it to other links. However, the transfer of information could lead to high bandwidth loads on other links. To overcome this problem, this paper proposes a method called "Avoid Passing High Utilization Bandwidth (APHUB)," which aims to (1) prevent the unloaded data putting additional load on the links when passing through the high bandwidth and (2) find a suitable new path. A comparison of the maximum bandwidth utilization using the proposed method with other algorithms showed that this method consistently produced the smallest bandwidth utilization; we thus consider it a better mitigation method than those presented previously.

Keywords: Software defined network · Offload · Path selection
Maximum utilization

1 Introduction

With the development of modern networks, software defined network (SDN) technology is gradually being developed to make it easier and more convenient for managers to control and operate the networks. However, security problems in the network are growing so fast that the design and updates for the defense strategies can hardly keep up due to the physical limits of the security devices. As a result, excessive congestion occurs in some links because of DoS attacks, leading to the loss of data packets and other problems. In the worst case, hackers might use this defect to cause a transmission inefficiency in the entire network. To avoid this problem, many studies have developed various Load-Balancing methods.

To avoid high bandwidth utilization caused by overuse in the same link, Load-Balancing methods transfer data by passing links with high bandwidth utilization to links with lower bandwidth utilization. This method is used mainly in decentralized IP Networks. The disadvantage of this method is that it cannot ensure that, after transferring to the links with lower bandwidth utilization, the next switch still has links with low bandwidth utilization available.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 70–78, 2018.
https://doi.org/10.1007/978-3-319-76451-1_7

One Load-Balanced method uses SDNs to monitor the switches and collect information about the usage of all links in the entire network so that it can calculate and reallocate the entire network bandwidth usage between all links. The new balanced load is achieved based on the results of the calculation. Though the two methods mentioned above solve the basic problem of high bandwidth utilization, they cannot ensure that, in the path from the source to the destination, the data always passes through the links that have the smallest bandwidth utilization. Therefore, this study aims to find a path that ensures the smallest maximum bandwidth utilization ($\hat{u}$) in all paths which is demonstrated by the following formula:

$$\begin{gathered} \hat{u} = Max\left\{P_i[U]\right\}, \mathrm{i} = 1, 2 \ldots, \mathrm{n} \\ \text{Find the P which } \hat{u} \text{ is minimum } \rightarrow \text{ Min } \hat{u} \end{gathered} \tag{1}$$

where P is the path, û is the maximum bandwidth utilization, i is one of the possible paths from source to destination, and $P_i[U]$ is the set of bandwidth loads for each link between the switches selected by this path ($P_i[U] = \left[\mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_\mathrm{m}\right]$).

The remainder of this article is structured as follows: The following section reviews previous literature to introduce existing flow control methods and compares them with the method proposed in this study. The third section presents the proposed method with an example of its operation. The fourth section lists the experimental results with the analysis and explanations, and the fifth section presents a summary of this article and discusses future developments.

2 Related Work

In 2016, Benjamin Baron et al. published an article [1], which used vehicle traffic for the centralized control of quality data unloading and used existing roads and road networks instead of data networks to mitigate the delays in traffic flow. This system takes advantage of the mobility of the vehicle by transmitting delay-tolerant traffic through the road network and uses the daily routine of the vehicle to reduce the traffic burden of the traditional data network. The author proposed the SDN-based structure in which a controller and a set of fixed wireless data storage devices make up the unloading point and are used as a forwarding engine. The controller receives the request to offload all or part of the data transmission and selects the vehicle flow for a series of offloading points that meet the transmission performance requirements in terms of bandwidth and latency. The controller solves the traffic distribution problem using Max-Min fairness allocation, calculates the unloading point sequence, connects to the unloading point, installs the forwarding state, and configures the scheduling policy.

As the amounts of users and data traffic on mobile networks has grown in recent years, the 3G/4G Base Station or Access Point Network service speed have declined, resulting in low QoS. To solve this problem, Jang and Chang [2] proposed a new approach called "Flow Management on Mobile Data Using SDN (FMSDN)", which uses software to define the characteristics of the network, depending on the different circumstances, to manage the flow and control of data, and compare the methods to the

Enforced Handover and Horizontal Handover theories. The purpose of this method is to prevent information loss from BS or AP, low QoS, or other issues caused by the overly huge data flow. However, the question remains whether a switch bandwidth can bear the huge amount of data in the huge information flow when passing from BS or AP down to SDN.

In the dynamic load balanced, the lack of strict routing synchronization increases the tendency of transient loops. In 2016, Li et al. [3] attempted to make the router achieve a dynamic load balance while avoiding the occurrence of transient loops in the IP network using two methods: Local Traffic Rerouting (LTR) and Global Traffic Rerouting (GTR). The only difference between the two methods is “Local” and “Global”; the core algorithms are the same. The focus is to prevent the link bandwidth usage from getting too large and leading to network congestion and packet loss. The use of LTR or GTR can dynamically adjust the flow of information and achieve a balanced load. In the algorithm, the way the links are connected to use the directed acyclic graph (DAG) ensures that the transient loops are avoided. In the path search, the search takes the shortest path, and the selection of the next node is relaxed. The threshold is used to dynamically adjust the path selection.

Lan et al. [4] proposed an algorithm called Dynamic Load-balanced Path Optimization (DLPO) which contains two main stages. The first stage is the initialization of the path in which the DLPO attempts to find a temporary path based on the available bandwidth of the bottleneck link for each path, and in all possible paths between the source and target hosts. The path with the largest available bandwidth on the bottleneck link will be selected as a temporary path. The second stage is the optimization of the dynamic path in which the DLPO changes the traffic path during traffic transmission to balance the link utilization and solve the congestion problem in the data center network. The DLPO load balancing consists of two algorithms: the Multi-link DLPO algorithm and the Single-link DLPO algorithm. The Multi-link DLPO algorithm can quickly balance the link utilization in the network to address some congestion paths, and the Single-link DLPO algorithm can reroute the traffic to avoid using high utilization links to solve the congestion path that the Multi-link DLPO algorithm cannot handle.

3 Approach

In this article, we propose an algorithm called “APHBU,” which collects current information, controls the status of the switch, and calculates an appropriate path to offload data. This method follows four steps:

1. Detection: The algorithm first detects whether the original path has a link with a high bandwidth utilization. The switch notifies the control layer, which recalculates the new path and passes it back to the switch.
2. Sort: The algorithm recalculates the path and uses the features of the SDN to collect the information downstream of the switch, mainly for each link load conditions, and to sort all of the links from small to large. This sorting procedure is used to create the Order List (L) in preparation for the next algorithm.

3. Connection path establishment: The algorithm defines a Select List (S), which starts with NULL; through the L, after DELETE the load link one by one from the smallest, add INSERT it into the S, and determine whether a path of S can be found between the source and the destination; if not, repeat the action of DELETE and INSERT to re-judge, until successful. By connecting the S path, the connected tree is formed.
4. Ensure the minimum bandwidth utilization in the path: After forming the connected tree, there may be plural paths between the source and the destination. To improve the path, the shortest path algorithm is applied to the connected tree.

3.1 Example

In this subsection, we take a basic network as an example, to compare the path found by APHUB with those found by other algorithms. First, as shown in the diagram of the network, Fig. 1, A is the source and H is the destination. The results of three algorithms—APHUB, Shortest Path Smoothing (SPS), and Minimum First Smoothing (MFS)—are compared.

1. APHUB
   First, it removes all links from the network and sort the links by size creating a list L = [Link(C, D), Link(D, F), Link(E, H), Link(A, B), Link(C, F), Link(A, C), Link(B, D), Link(D, H), Link(F, G), Link(F, H), Link(A, E), Link(C, G)]. Next, it joins the links one by one and decide whether A can be connected to H successfully. When the link comes to Step. 7, as shown in Fig. 2, the connection is successful, and the formed structure is recorded as a connected tree. A multiple path is found, and the shortest path is applied to the connected tree. The best path is found as A → C → D → H, and the maximum link utilization is the last link, namely, Link (D, H), whose value is 7.

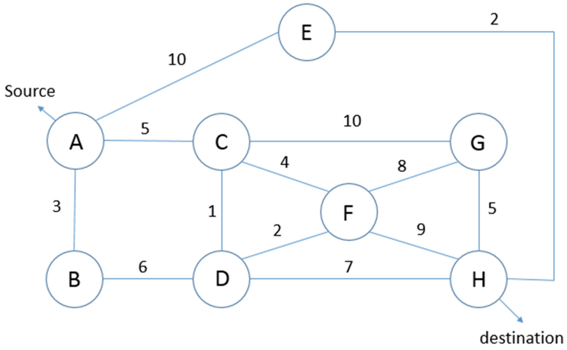


Fig. 1. Example network.

2. Shortest Path Smoothing (SPS)
   SPS is a typical path search algorithm, which selects the route by finding all the paths connecting A → H and by summing the utilizations in all paths. The path

with the smallest utilization is selected, as shown in Fig. 3. The selected path is A → E → H, but the maximum utilization is Link (A, E), whose value is 10.

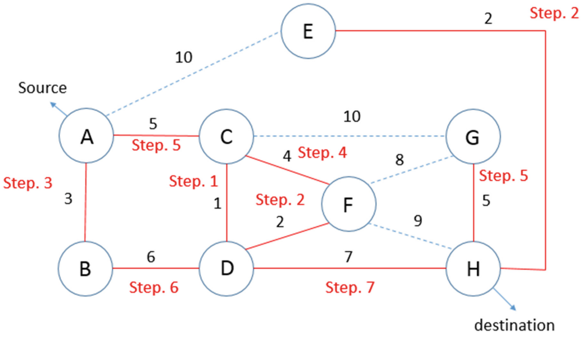


Fig. 2. APHUB outcomes.

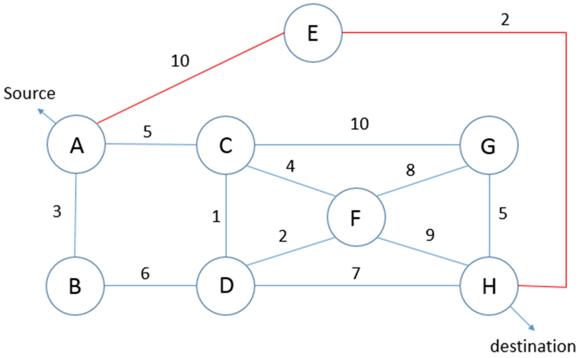


Fig. 3. SPS outcome.

3. Minimum First Smoothing (MFS)
   MFS is an intuitive algorithm that directly selects the minimum utilization of the link to forward; the time complexity is low and can quickly determine the path.

Step 1. Of the three links after A, the minimum bandwidth link, Link (A, B), is selected.

Step 2. After reaching B, as the connection cannot move backward, only one link is available, namely, Link (B, D), so the connection continues here.

Step 3. After reaching D, there are three links, and the minimum bandwidth link, (D, C), is selected.

Step 4. After reaching C, there are three links, the minimum bandwidth link, Link (C, F), is selected.

Step 5. After reaching F, there are three links, the minimum bandwidth link is (C, D). However, as D has already been visited, the second smallest Link (F, G) is selected.

Step 6. After reaching C, there are two links, the minimum bandwidth link, Link (G, H), is selected.

To arrive at the destination, the selected path is A → B → D → C → F → G → H. In the path, the maximum link bandwidth utilization is Link (F, G), whose value is 8 (as shown in Fig. 4).

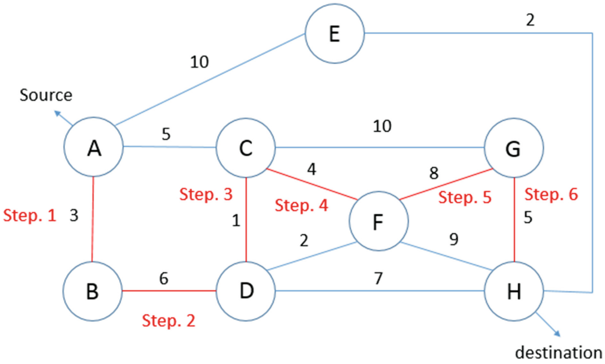


Fig. 4. Step-by-step diagram of MFS.

Consolidating the results of the three algorithms results in Table 1. The path found by APHUB has the minimum value of $\hat{\boldsymbol{u}}$.

Table 1. The results of the three algorithms.

Algorithm	Path	$\hat{u}$
APHUB	[Link (A, C), Link (C, D), Link (D, H), Link (G, H)]	Link (D, H) = 7
Shortest Path Smoothing (SPS)	[Link (A, E), Link (E, H)]	Link (A, E) = 10
Minimum First Smoothing (MFS)	[Link (A, B), Link (B, D), Link (D, C), Link (C, F), Link (F, G), Link (G, H)]	Link (F, G) = 8

4 Experiment

This section compares the algorithms presented in the examples to compare the simulations in terms of the maximum link utilization of each û and the average size of the bandwidth. To make the simulation easier, this article uses C language to perform the experiment. Table 2 shows the pre-set parameters before the experiment is carried out (Table 3).

Table 2. Experimental simulation environment

SW/HW	Description
Processor	Intel Core i7 3.40 GHz
RAM	4 GB
Operating System	Microsoft Windows 7
Programming Language	C Language

Table 3. Parameters in the experiments

Parameter	Quantity
Number of switches	100–500
Offload	5–35 (%)
Output Link	1–10
Utilization	0–99 (%)
Average bandwidth utilization in the network	30–70 (%)

In this lab, the assumed specification for each switch is the same. Output link and Bandwidth Utilization are different. The topology of the network is a random connection, and the value of Bandwidth Utilization is given by the Gaussian discrete method, so that the value belongs to the normal distribution. In this experiment, we discuss how different averages will lead to the relation between the average and the maximum Bandwidth Utilization. In the experiment, three path search methods, APHBU, MFS, and SPS are discussed.

The relation between the maximum link bandwidth load rate and the average bandwidth of the entire network link bandwidth is shown in Fig. 5. It can be seen from the figure that APHUB can effectively keep the maximum link bandwidth utilization in the path below the average value of the network link bandwidth utilization. For MFS and SPS, the situation is as follows:

1. As mentioned above, the maximum utilization of the MFS maximum load is the largest among the three, much larger than the average, or is unable to let the data offload connect to the link with maximum utilization, thus resulting in failure.
2. The maximum utilization of the SPS is significantly smaller compared to that of MFS, but the maximum link bandwidth utilization of its path is still not comparable to that of APHUB, and the minimum value cannot be achieved.

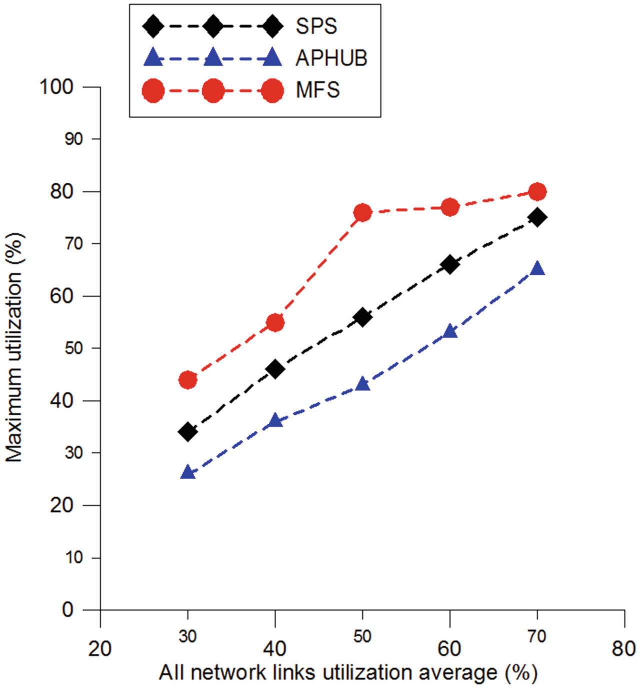


Fig. 5. The relation between the average bandwidth utilization and the maximum bandwidth utilization.

5 Conclusion

The purpose of this paper is to prevent links to high bandwidth utilization when the data path is reproduced and to ensure that the path is optimized so that the highest link utilization is the smallest of all paths. In addition, the new data path can ensure that all links in the path of the data flow are low load links, thus balancing the load. As the link in the path of the largest utilization joins last, it can be clearly and quickly known which link requires further analysis.

Acknowledgement. This research is partially supported by Ministry of Science and Technology, Taiwan under Grant no. MOST 105-2221-E-194 -014 -MY2 and MOST 106-3114-E-006-003.

References

1. Baron, B., Spathis, P., Rivano, H., et al.: Centrally controlled mass data offloading using vehicular traffic. IEEE Trans. Netw. Serv. Manage. **14**(2), 401–415 (2017)
2. Jang, H.-C., Chang, C.-H.: Context aware mobile data offload using SDN. In: 26th International Telecommunication Networks and Applications Conference (ITNAC) (2016)
3. Li, K.-Y., Chen, C.-W., Lee, S.W.: Dynamic load balanced routing in IP networks. In: 6th International Conference on Information Communication and Management (2016)
4. Lan, Y.-L., Wang, K., Hsu, Y.-H.: Dynamic load-balanced path optimization in SDN-based data center networks. In: 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP) (2016)

An Extension of Attack Trees

Yi-Chih Kao[1], Yuan-Ping Hwang[2], Shih-Chen Wang[2], and Sheng-Lung Peng[2(✉)]

[1] Information Technology and Service Center, National Chiao Tung University, Hsinchu, Taiwan
ykao@mail.nctu.edu.tw

[2] Department of Computer Science and Information Engineering, National Dong Hwa University, Hualien, Taiwan
{810513102,810621002,slpeng}@gms.ndhu.edu.tw

Abstract. Attack trees provide a model to describe the security of a system based on the possibility of various attacks. In this paper, we propose the concept of “attack graphs” as an extension of attack trees, wherein directed acyclic graphs are used to depict possible attacks on a system. By deploying this model, system managers can discern all possible threats to the system and thus are more likely to design efficient countermeasures to thwart those attacks. Within this model, we also propose the concept of the most dangerous path in the attack graph, and finally propose an algorithm to expose it.

Keywords: Attack trees · Directed acyclic graph · Attack graph
The most dangerous path

1 Introduction

In the past few years, ubiquitous digitalization and widespread use of the internet have been accompanied by increasing cyber-attacks and intrusions while the technological defense methods reciprocally prosper and become treacherously complicated. In fact, the reason why attacks can break through existing security defenses is often because they are able to dodge security methods in ways that designers have not thought of. Undoubtedly, a threat modeling method for computer system security is in high demand. If we can predict various attack methods and simulate solutions before systems are attacked, we stand a fair chance to prevent possible attacks. In addition, if we can identify the attackers, understand their motives and targets, and be aware of the security capabilities of computers, we may be able to build appropriate solutions that deal with real threats.

This study presents a graph-based model that describes attacks and predicts a variety of different intrusion paths based on the systematic thinking of security management organizations. Thereby, it provides a fast, specific, and organized method for examining system security strategies and adjusts them to achieve the ultimate purpose of attack prevention and enable their alignment with security situations [1].

System engineers have for a long time relied on failure mode and effect analysis (FMEA) to improve their system security designs by analyzing failure data and types of failure [2]. The main purpose of this approach is to identify risks and enable control

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 79–85, 2018.
https://doi.org/10.1007/978-3-319-76451-1_8

mechanisms or minimize the identified risk factors. Recognition of risks enables the creation of more realistic project plans. The FMEA technology helps identify failure potentials and enables the detection of problems in the early warning processes before they occur [3].

Engineers improve their system designs based on failure data in historical context. Both software engineers and information system administrators do the same. However, IT professionals for system operations security usually utilize attack information instead of failure data to improve the security of computer network systems and some of the components connected to these systems. The reasons are organizational changes such as fatigue and discontent with the leak of information (fear of a decrease in public trust, etc.), as well as the preference to employ resistance against attacks based on detailed and reliable attack data [4].

Despite that the fatigue of some organizations has been exposed on their systems, attack data have become readily available in the last few years. This is because mass communication tools and media have been paying more attention to information security issues and other resources such as the SANS Internet Storm Center [5] and SecurityFocus [6], as well as many Computer Emergency Readiness Teams and Coordination Centers, which have been consolidating real-time and comprehensive security-related information. These resources announce security vulnerabilities, discuss their nature in detail, and describe how to utilize and fix them.

Several studies have described the use of attack models for testing the security levels of enterprise systems. The authors of these publications argue that understanding a system requires the use of different methods and models, wherein attackers can actually help IT professionals who are responsible for the system design to achieve reliable security systems that hinder their attacks. In addition, these IT professionals can present appropriate solutions that deal with real threats if they can understand who these attackers are, recognize their attacking capabilities, and discern their motives and targets [1].

Basically, understanding the constructs of threat models can help developers and integrators build robust and reliable systems. In early studies, researchers predominantly used a tree structure, i.e., the so-called attack tree, to deduce possible threats and describe their systems. For example, after a website has been built, an administrator can easily use a threat model to test it. The same applies to IT professionals as they try to envision vulnerability in enterprise information assets. By developing several attack trees (thus forming an enterprise attack forest), IT professionals can obtain evaluation accreditations for hundreds of potential vulnerabilities in an enterprise system, and thereby improve the security of their computer network. Several organizations, including general budget, accounting and statistics offices, as well as the U.S. Department of Homeland Security, have engaged Red Teams to identify vulnerabilities in their information assets [7].

However, the construction of system attacks that is described in attack trees is too narrow. In attack trees, there is only one path by which the attack can lead to the base of the system (root). Therefore, many circumstances cannot be described using attack trees. In order to solve this problem, we allude to the concept of "attack graphs" for an attack that is portrayed as a node, but it belongs to many different paths. In this way, we can describe the attack in a more realistic way. In addition, the danger in each route can be defined and therefore the concept of "the most dangerous path" can be realized, which would assist system designers to enhance the defense and avoid severe damage.

This paper comprises six sections. Section 2 will introduce studies related to attack trees. Section 3 leads to the concept of attack graphs. Section 4 presents the concept of the most dangerous paths and delivers an algorithm that identifies the most dangerous path within an attack graph. The final section presents the conclusions of this study.

2 Attack Trees

A tree can be defined in several ways, and recursion is a natural way to describe it. A tree is the collection of all its nodes, which may be empty and thus sometimes be marked as Ø. If this is not the case, then a tree is composed of a special node r called the root, and zero or more subtrees T_1, T_2, …, T_k. The root of each subtree has a directed edge that points into node r and can be described as a child of the latter. According to the recursive definition [8], an n-node tree has a special node called the *root* and n-1 edges, where each node except for the root has a parent node, as shown in Fig. 1.

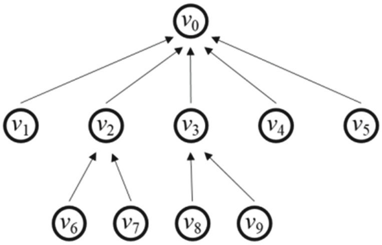


Fig. 1. A tree.

In the tree shown in Fig. 1, the root is v_0 and nodes v_1 to v_5 are children of v_0 while v_3 is the parent of v_8 and v_9. Each node can have any number of children, which may also be zero. A node without children is called a *leaf*, which is the case for v_1, v_4, v_5, v_6, v_7, v_8, and v_9 in Fig. 1. Nodes that share the same parent are called *siblings* and therefore v_1 to v_5 are all siblings. The relationships between grandparents and grandchildren can also be defined in the similar way.

In such a tree, there is a single path from any node to the root node. In this path, the number of edges defines the length of the path. The longest path therefore defines the height of the tree [8, 9].

Attack trees are conceptual diagrams that can portray the attack. A complete attack tree may contain hundreds of thousands of different attacks paths. Even so, these trees are very useful when determining which threats exist and how to deal with them. Attack trees can lend themselves to defining an information assurance strategy [1, 2].

How do we create an attack tree? First, identify the possible attack goals. Each goal forms one single tree (although they may share subtrees and nodes). Second, think of all the attacks that would lead to the goal, and then add them to the tree. Repeat the process. Once you finish, you can use it under any circumstances. Leaves represent the attacks, and nodes represent the targets. The root usually represents the whole system. In an attack tree, there are AND nodes and OR nodes. OR nodes are alternatives. For example, there are four ways to open a safe. AND nodes represent different steps toward achieving the same goal. Attackers cannot achieve the goal unless both subgoals are satisfied. This is the basic attack tree [1].

For example, we show how the attack tree is used in the actual situation. The attack tree portrayed in Fig. 2 demonstrates a possible computer virus attack [1, 2].

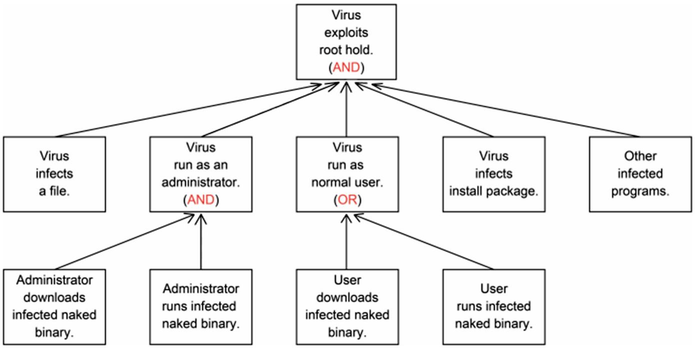

Virus exploits root hold. (AND)
Virus infects a file.
Virus run as an administrator. (AND)
Virus run as normal user. (OR)
Virus infects install package.
Other infected programs.
Administrator downloads infected naked binary.
Administrator runs infected naked binary.
User downloads infected naked binary.
User runs infected naked binary.

Fig. 2. An attack tree for computer virus attack.

3 Graph Model: Attack Graphs

Let $G = (V, E)$ be a graph where V is the node set and $E \subseteq V^2$ is the edge set. For other graph-related definitions, we refer to [10, 11].

The graph $G = (V, E)$ is a directed graph if (u, v) is an edge, then the edge is from vertex u to vertex v with v commonly known as the head, and u as the tail of the edge. In a directed graph, the (u, v) edge differs from a (v, u) edge (unlike in a typical undirected graph, both are the same).

In directed graphs, on the other hand, there is a direction to follow, such a path leads in one direction only. The beginning of the path is called the starting point, while the end is called the finish point. When the path starts from the starting point and ends at the same position, which indicates the starting point is equal to the finish point [10], this path is called a cyclic graph. If no cyclic path exists in a directed graph, then the graph is called a DAG (Directed Acyclic Graph). The DAG is often used to demonstrate certain problems in computer [12].

The concept of the attack graph comes from the attack tree. In the attack tree, every leaf represents an attack, while nodes are like forts wherein AND or OR nodes stand for the defense capabilities. To break through an AND node, all its subnodes have to be broken through as well. However, to break through an OR node, one only requires one of its subnodes to be accomplished. Every leaf has only one way that leads to the root, which usually represents a system headquarter. The entire system will shut down once the headquarter fails.

If a single attack can harm more than two forts, it cannot be characterized by the attack tree. To describe an alternative, we came up with the concept of the attack graphs. It is a DAG. Attack graphs no longer follow the tree structure.

Building an attack graph involves two steps. First, one needs to break down a system from top to bottom hierarchically. The resultant breakdown usually results in a tree structure with the root node as the entire system. Using a company as an example, this is usually a typical organizational hierarchy chart. By including the direction from a child to the parent on each edge in the tree, a DAG is formed.

Second, each possible attack needs to be presented as a node. If node u can attack v, then we build a directed edge directing from u to v. When all nodes have been considered, an attack graph is created.

In our attack graphs, apart from the starting point (sources), all nodes have AND or OR node indicators. All the attacks pointing to an AND node have to be satisfied, while the OR node only needs one of the conditions. We can add AND/OR nodes to complete the full structure of attack tree. Figure 3 is an attack graph with indicators for a computer virus infection system.

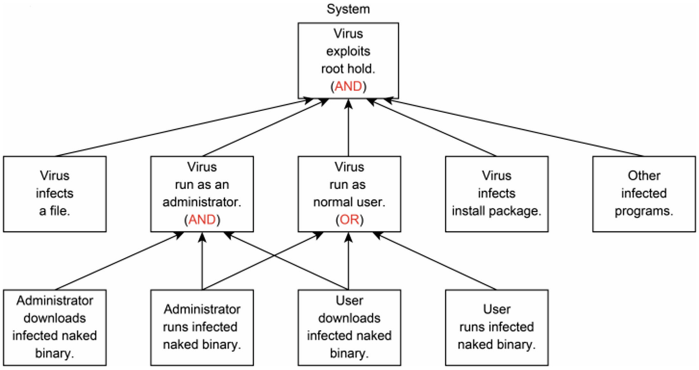

System
Virus exploits root hold. (AND)
Virus infects a file.
Virus run as an administrator. (AND)
Virus run as normal user. (OR)
Virus infects install package.
Other infected programs.
Administrator downloads infected naked binary.
Administrator runs infected naked binary.
User downloads infected naked binary.
User runs infected naked binary.

Fig. 3. An attack graph for computer virus attack.

4 The Most Dangerous Path

If we appoint an initial attack probability to every source (in-degree zero) and indicate all other nodes as AND/OR nodes, we can define the system's attack probability as follows:

- u is an AND node: Assume that vertices in $\{u_1, u_2, \ldots, u_d\}$ are its lead nodes, which means (u_i, u) defines an edge in the graph. Let the probability of u_i be p_i. Then $\prod p_i$ is the probability of u to denote the success rate of all u_i being attacked.
- u is an OR node: Assume that vertices in $\{u_1, u_2, \ldots, u_d\}$ are its lead nodes, which means (u_i, u) defines an edge in the graph. Let the probability of u_i be p_i. Then $\sum p_i$ is the probability of u to denote the success rate of all u_i being attacked.

With the abovementioned definitions, the attack graph becomes a DAG. Every node from the starting point to the target has an attack probability. The node that has the lowest probability defines the rate of a successful attack. This rate could be defined as the barrier for this attack path. Therefore, when we call a path "the most dangerous path," it means it is most likely to be broken through.

Determining the most dangerous path in an attack graph helps the administrator to come up with appropriate countermeasures or necessary safety precautions to block the attack. We can calculate the most dangerous path by following the method below. For this, we take the assumption that all attack probabilities are the same at all attacking points:

First, we add up the amount of attack sources. Let us assume there are k of them. Thus, the probability of each source is $1/k$. We work in the following topological ordering:

- If node u is an AND node, then its probability is $\prod p_i$, where p_i is the probability of its predecessor u_i for some i.
- If node u is an OR node, then its probability is $\sum p_i$, where p_i is the probability of its predecessor u_i for some i.

We mark the target nodes first, then scan in reverse order according to the topological ordering. We mark the predecessor with the highest attack probability, and repeat the steps until we meet the source. We connect all the nodes marked to form a path, which is the most dangerous path we are looking for. According to the definition of the most dangerous path, we can easily prove that the path we just found out using the algorithm is the most dangerous path. The following is our algorithm:

```
 Algorithm MDP
 Input: an attack graph
 Output: the most dangerous path
1: let $k$ be the number of source vertices
2: for each source vertex $v$, $f(v) = 1/k$
3: from source vertices to sink vertex do
     $v$ is OR: $f(v) = \sum f(u_i)$ where $u_i$ is a predecessor vertex of $v$
     $v$ is AND: $f(v) = \prod f(u_i)$ where $u_i$ is a predecessor vertex of $v$
4: let $u$ be the sink vertex
5: while $u$ is not a source vertex
     let $v$ be a predecessor vertex of $u$ with maximum $f(v)$
     let $u = v$ and mark $u$
6: output the path induced by marked vertices
End of Algorithm
```

Finally, we analyze the time complexity of this algorithm. Steps 1, 2 and 4 are the initial statements, which the settings could be completed within O(|V|). Steps 3 and 5 are loops that require O(|V| + |E|), and therefore the total time consumed is O(|V| + |E|).

It is worth mentioning that the probability for each attack source could be different. This case would only require a small modification in Steps 1 and 2.

5 Conclusion

When an attacker enters the system, the attack tree has only one way to describe the attack, *i.e.*, by following the tree structure. The attack graph describes the attack in a more diverse way, *i.e.*, a single attack point can belong to a variety of attack paths.

In this study, we come up with the concept of the attack graphs as an extension of the attack trees. Taking attack probabilities as the parameter, the idea of the "most dangerous path" is introduced. A linear formula is presented to calculate the most dangerous path.

In conclusion, the present study predominantly focuses on how to modify the attack trees to attack graphs and thus produce warnings exhibiting the weak links by calculating the most dangerous path.

Acknowledgement. This work was partially supported by the Ministry of Science and Technology of Taiwan, under Contract No. MOST 106-2221-E-259-005.

References

1. Schneier, B.: Attack trees. Dr. Dobb's J. **24**(12), 21–29 (1999)
2. Odubiyi, J.B., O'Brien, C.W.: Information security attack tree modeling. In: Proceedings of Seventh Workshop on Education in Computer Security (WECS), pp. 29–37 (2006)
3. Shooman, M.L.: Probabilistic Reliability: An Engineering Approach. McGraw-Hill Book Company, New York (1968)
4. Anderson, R.: Why cryptosystems fail. In: Proceedings of the 1st ACM Conference on Computer and Communications Security (1993)
5. SANS Internet Storm Center. http://isc.sans.org
6. Security Focus. http://www.securityfocus.org
7. Ray, H.T., Vemuri, R., Kantubhukta, H.R.: Toward an automated attack model for red teams. IEEE Secur. Priv. **3**(4), 18–24 (2005)
8. Horowitz, E., Sahni, S., Mehta, D.P.: Fundamentals of Data Structures in C++, 2nd edn. Silicon Press, New York (2007)
9. Weiss, M.A.: Data Structures and Algorithm Analysis in C, 3rd edn. (2007)
10. Diestel, R.: Graph Theory. Springer, Heidelberg (2005)
11. West, D.B.: Introduction to Graph Theory. Prentic-Hall Inc., Upper Saddle River (2001)
12. Skiena, S.S.: The Algorithm Design Manual, 2nd edn. Springer-Verlag, London (2008)

Face Detection in a Complex Background Using Cascaded Conventional Networks

Jianjun Li[1], Juxian Wang[1], Chin-Chen Chang[2](✉), Zhuo Tang[3], and Zhenxing Luo[3]

[1] School of Computer Science and Engineering, Hangzhou Dianzi University, Hangzhou, China
lijjcan@gmail.com

[2] Department of Information Engineering and Computer Science, Feng Chia University, Taichung, Taiwan
alan3c@gmail.com

[3] Key Lab of the 36th Institute of CETC of China, Jiaxing, China

Abstract. Although significant achievements have been achieved in the field of face detection recently, face detection under complex background is still a challenge issue. Especially, face detection has wide applications in real life, such as face recognition attendance system and crowd size estimation. In this paper, we propose a novel cascaded framework to tackle the challenges based on: blur, illumination, pose, expression and occlusion. Our framework adopt the localization of facial landmarks to boost up their performance. In addition, our detector extracts features from different layers of a deep residual network for complementary information of low-dimensional and high-dimensional features. Our method achieves notable results over the state-of-the-art techniques on the challenging WIDER FACE benchmark for face detection and our results show that average precision of 89.2%. Importantly, we demonstrate superior performance and robustness in a challenging environment.

Keywords: Face detection · Cascaded conventional neural network
Facial landmarks

1 Introduction

Face detection in the field of computer vision has an unusual position and it also has widely research value in artificial intelligence interaction, video conferencing, identification of identity, vehicle safety and so on. Face detection is one of the visual tasks that humans can do effortlessly. However, in computer vision terms, this task is not easy.

Numerous techniques of the current face recognition assume that the availability of frontal faces of similar sizes. The background in these images is eliminated so that the accuracy of face detection is higher. However, in realistic application scenarios, this hypothesis may not set up because of the facial appearance and environment conditions. Faces may occur in a complex background and in many different postures, the detection result will vary under different illumination conditions (such as day and night, indoor and outdoor), different occlusion degree of human face (such as masks, sun-glasses, hair, beard etc.), different age groups, unconstrained pose and other factors. The above

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 86–97, 2018.
https://doi.org/10.1007/978-3-319-76451-1_9

different factors caused that detection system are prone to detect certain areas of the background as human faces by mistake, or failed to detection real faces.

In order to rectify the problem, a series of detectors proposed by many researchers are designed to cope with these problems. Farfade et al. [1] did significant research on the problem of multi-view face detection, and their method also can handle occlusion to some extent. In their work, a deep dense face detector (DDFD) they addressed uses a single model based on deep convolution neural network to detect faces in a wide range of orientations, without pose/landmark annotation. To improve the accuracy for the face recognition system under the variant light conditions. Tran et al. [2] proposes a new approach that the illumination of each test image is adjusted by singular value decomposition (SVD) of the training images before the features are extracted to solve illumination variation. Zhang et al. [3] proposed a multi-task cascaded networks framework for face detection in unconstrained environment, which marks a step forward in face detection.

As algorithms proposed above, most of these algorithms are designed to solve certain problems. So the key issue is: Can we propose a model to address a comprehensive problem? In this paper, our network architecture effectively solves these complex issues of heavy blur, overlap, extreme illumination, small objects and irregular posture etc. Our framework is integrated with cascaded convolutional neural networks (CNNs), which is designed from two aspects. The first network aims at detecting faces through a deep residual neural network (ResNet) [4], and then eliminates the excess bounding boxes by non-maximum suppression (NMS). In the second network, it will produce facial landmarks location based on the output of the previous network through a more lightweight network, and use these attributes of facial landmarks to exclude the wrong boxes with higher performances.

On the whole, the contributions of this paper are mainly summarized in the following two aspects:

1. We propose a new cascaded CNNs based on framework for face detection under complex background. Extensive experiments are conducted on challenging benchmarks, to show superior performance than the state-of-the-art methods and has its advantages of simplicity and robustness.
2. We explore the relationship between face detection method and the facial landmarks location. Experimental results show that facial landmarks contribute to enhance the accuracy of detection. It is equivalent to a verification procedure and reduce false positive rate by dint of the relationship.

The rest of the paper is organized as follows: In Sect. 2, we briefly introduce related research in the area of face detection. Section 3 presents the framework of our work and provides a more detailed description for each modules. Section 4 provides the experimental results, and conclusions are in Sect. 5.

2 Related Work

2.1 Multi-scale Representation

Multi-scale representation is very important in image processing. Lowe [5] proposed a scale invariant feature transform algorithm (SIFT) that keeps the invariance of image translation, rotation, zoom, and affine transformation. Speed up robust features algorithm (SURF) proposed by Bay [6] is similar to the above SIFT method. In Ramanan et al.'s work [7], their algorithm mainly uses multi-scale representation and deformable part model for object detection and their result outperforms the best results in the 2007 challenge. Researchers have shifted from visual geometric restoration to more object recognition problems since the emergence of Bag of Words (BOW), spatial pyramids and vector quantization. Single Shot Multi-Box Detector (SSD) [8] is the recent technique that obtains predicting category scores and a series of fixed-size bounding boxes. This network is different from the base network, but add additional auxiliary structure. However, all these multi-scale representations describe local features of images in a simple form at different scales. Our work aims to choose the appropriate scale invariant method to optimize our model so that our model can be better applied to multi-scale faces detections.

2.2 Face Detection

Face detection is the key step of face recognition and also an indispensable part of face application. However, it also meets with many challenges, such as blur, occlusion, extreme lighting, and large pose variation, in real applications.

Early detection methods based on geometric features [9–12] have the characteristics of small storage and immunity to illumination interference, but require high quality of image and high accuracy of feature points. Now it is often used as an aid to other methods of detection. Compared with these traditional methods [13, 14], neural network method has its unique advantages in face detection and recognition. When implementing with GPU, it can significantly improve detection speed. In recent years, more complex networks were applied to the face detection, such as VGGNet [15], GoogleNet [16], ResNet, etc.

Facial landmarks location is not only a key problem in face recognition research field, but also a basic problem in the field of graphics and computer vision. The purpose is to find landmark locations on the images that correspond to facial features such as the eyes, nose, and mouth. The basic idea of the traditional location algorithm, such as active shape model (ASM) [17] and active appearance model (AAM) [18], is that combine the texture features of faces with the position constraints among the feature points. Certainly, there are template fitting approaches as well, such as the methods of [19–21]. Recently, Zhang et al. [22] use the inherent correlation between face detection and face alignment to boost up the performance.

However, most of the available face detection and face alignment methods has not exploited the inherent correlation between these two tasks. We investigate to associate with facial landmarks location and face detection.

3 The Proposed Approach

In this section, we will depict the architecture of cascaded networks for face detection and describe the characteristic of each proposed approach in detail.

3.1 Overall Framework

The overall framework is divided into three parts for different purposes as shown in Fig. 1. The input of our approach could be an arbitrary picture and the final bounding boxes are disposed by NMS.

Stage 1: We initially resize the input image to different scales to build a coarse image pyramid, the size of scale factors could be set according to a certain proportion. Our method uses the bilinear interpolation method to downsample the input images.

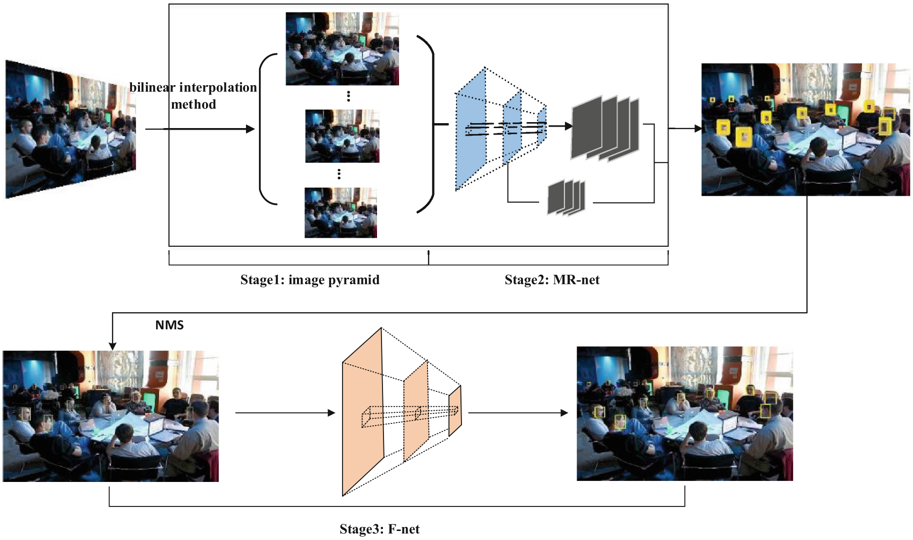


Fig. 1. Overview of our cascaded networks that includes three stages. The yellow boxes represent the result of detection.

Stage 2: Multi-scale images generated by image pyramid are fed to the first network for face detection. The characteristic of detection network is able to extract high-dimension features, namely response maps, using modified 101-layers residual network (MR-net). These response maps are extracted by end to end from different layers. Based on the response maps, we obtain detection bounding boxes for each scale and then merge them back in an original scale. After that, the final detection bounding boxes will be extracted by employing the NMS.

Stage 3: This stage uses a facial attribute as an auxiliary task to enhance face detection performance by exploiting another convolutional neural network (F-net). Then, this network outputs five facial landmarks to verify whether the face is true or not, if the facial landmarks cannot be detected, these boxes from MR-net are excluded. The advantage of this network is that it reduces false positive rate. The final result is shown as Table 1.

Table 1. Performance of our approach on validation set of WIDER FACE. Underline indicates the best performance.

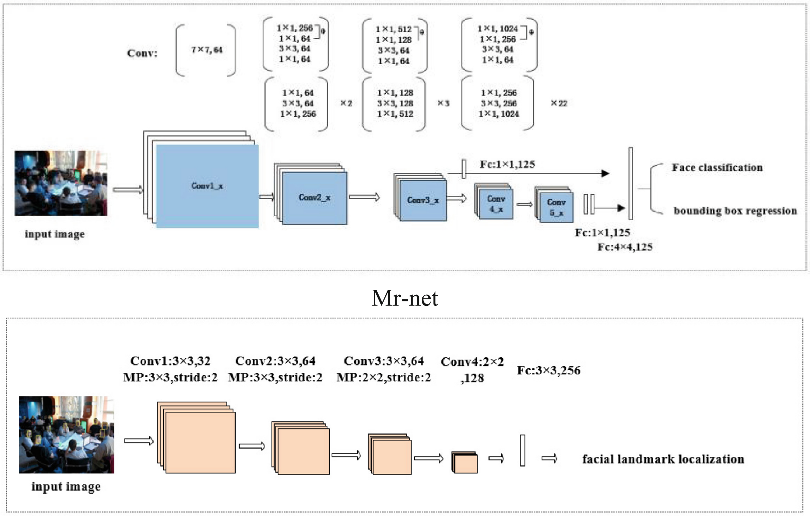
Method	Easy	Medium	Hard
ACF [23]	0.659	0.541	0.273
Two-stage CNN [24]	0.681	0.618	0.323
Multi-scale cascade CNN [25]	0.691	0.634	0.345
LDCF+[26]	0.790	0.769	0.522
Multi-task cascade CNN [3]	0.848	0.825	0.598
Ours	0.892	0.863	0.702

3.2 Design of Cascade Network Architecture

Each network of our cascade network has its own characteristics. The MR-net is used to detect accurate faces in the wild, this network we design has good robustness and detection effect in the complex background, the results can be seen in the experimental chapter. The structure of MR-net is a modification of the residual network, not just extracting the last layer feature for classification. However, we takes more into account the features from other layers and then merge with the features of the last layer. The advantage of MR-net for face detection as the following:

1. MR-net network can solve the problem of gradient disappearance, it increases the depth of networks, but can ensure the best accuracy at the same time. Because the classification operation is not performed only at the last level, this operation is similar to google-net.
2. We modify the network based on 101-residual network to produce multi-results from different layers by end-to-end detection and then merge them. You can also think MT-net as a model Pyramid, namely the output of the lower layer is equivalent to the result that a small model works on the image, and the output of the last layer is equivalent to the result that the large model works on the image, then merge them as the final result.

The last network of our cascade network is F-net, the function this network assist the results of detection with more accurately. Namely we add check step on the output of MR-net, this step allows face detection to be more accurate. Our cascade network architecture is showed in Fig. 2.



Mr-net

F-net

Fig. 2. The architectures of MR-net and F-net, where “Fc” indicates fully connected layers, “MP” indicates max pooling and “Conv” indicates convolution.

3.3 Training

For training MR-net, we define the class label (positive or negative samples) to each object: (i) Positives: Regions that the intersection-over-union (IOU) overlap higher than 70% with any ground-truth boxes; (ii) Negatives: we assign a negative label to a non-face background if its IOU overlap is lower than 30% with any ground-truth boxes (others are ignored). With these definitions, we minimize an objective function. We use log loss function for face classification and Huber loss for bounding box regression for each sample xi. Obviously, we need a multi-task loss and it is defined as

$$L(p_i, r_i) = \frac{1}{N_{cls}} \sum_i L_{cls}\left(p_i, y_i^{gt}\right) + \lambda \frac{1}{N_{reg}} \sum_i p_i L_{reg}\left(r_i, r_i^{gt}\right) \tag{1}$$

Here i is the index of object in mini-batch and $\mathrm{p_i}$ is the predicted probability of object i being a face. The notation $y_i^{gt} \in \{0, 1\}$ denotes the ground-truth label. $\mathrm{r_i}$ is a vector representing the 4 parameterized coordinates of the predicted bounding box, including left top, height and width. r_i^{gt} is the ground-truth box coordinates.

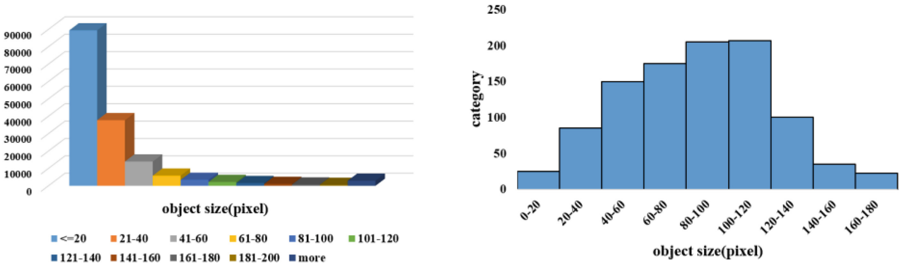
We implement our method using the caffe package [27] and used its pretrained ImageNet [28] model for fine-tuning on the WIDER FACE [24] training set. During training, we randomly crop the training images to 227 × 227 region and we define the learning rate of 10^{-5} and momentum of 0.9. Here the training data is WIDER FACE, which dataset consists of 393,703 labeled face bounding boxes in 32,203 images, where 50% of them for testing set consists of three subsets according to the hierarchy of images, 40% for training set and the remaining for validation set.

4 Experiments

In this section, we first analyze the distribution of data and amplify data if the data is unbalanced. Then we evaluate the detection performance against other methods and qualitative results on WIDER FACE. At last, we investigate the impact of using the localization of facial landmarks, which assists the face detection to reduce the false positive rate.

4.1 Data Analysis

We analyze the distribution of the average object scale on WIDER FACE. As shown in Fig. 3(a), we found that more than 78% faces had an average size between 8 and 40 pixels approximately. Smaller objects obviously exceed larger objects, and here exists the phenomenon of imbalanced data. Therefore, we add one step of the data augmentation. In the data amplification section, we use simple cropping, scaling and rotation. Certainly, it is necessary to consider the distribution of pre-trained dataset (ImageNet), as shown in Fig. 3(b). During the training phase, the data analysis is important to obtain a better training model.

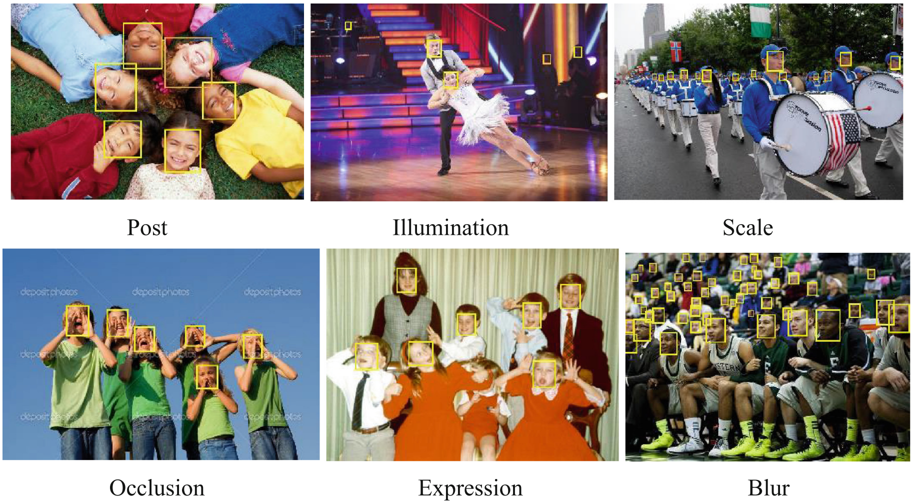


(a) (b)

Fig. 3. The statistic of object sizes on the training dataset. (a) The distribution of WIRED FACE dataset and different colors represent different ranges of face pixels. (b) The distribution of pre-trained ImageNet dataset.

4.2 Evaluation on Face Detection

In this section, we visualize qualitative result of our detectors in Fig. 4. We choose challenging samples with high occlusion, exaggerated expression and other cases (atypical pose, blur, illumination etc.). The results show that our method has both better robustness and higher accuracy. We include a detailed comparison for the following methods [3, 29], and datasets are divided into five groups according to attributes. As the experimental results shown in Table 2, our model works well in any case of challenging dataset, which proves that the characteristic of F-net play an important role as well.



Post

Illumination

Scale

Occlusion

Expression

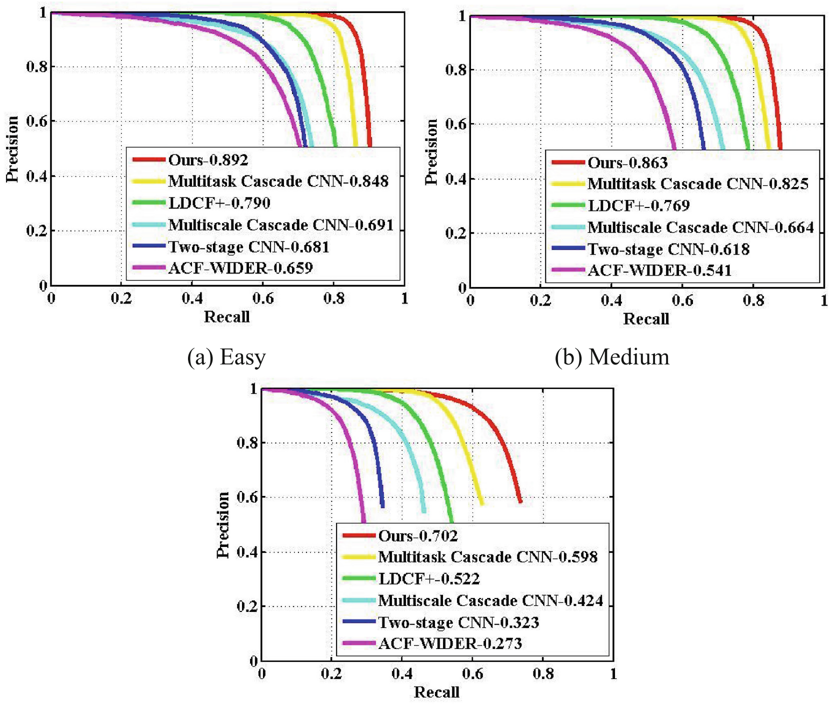
Blur

Fig. 4. Visualize results for each challenging situation with our method.

Table 2. Experimental result validate that our method has notable robustness for various situations. The seetaface [29] approach is an open source C++ face recognition engine.

Method	Blur	Occlusion	Expression	Illumination	Pose
Our	0.867	0.656	0.941	0.8351	0.899
MT-CNN [3]	0.840	0.603	0.933	0.7786	0.795
Seetaface [29]	0.5556	0.1906	0.9333	0.7421	0.8333

The performance of our proposed detector has been compared with other state-of-the-art face detectors [3, 23–26] when using the publicly available dataset of WIDER FACE. Our face detector is able to achieve excellent results on challenging dataset. Importantly, compare with multitask cascade CNN algorithms, the performance of ours improves by 10.4% on "hard" set. As shown in Fig. 5, our method obviously can improve precision and recall rate on "hard" and "easy" set respectively.



(a) Easy

(b) Medium

(c) Hard

Fig. 5. Precision recall curves on three subsets of WIDER FACE validation set.

4.3 The Effectiveness of Joint Face Detection and Facial Landmarks

We conduct related experiments to compare two different approaches (with and without the localization of facial landmarks) on WIDER FACE and evaluate the joint contribution of detection and facial landmarks. Qualitative result shows that it is beneficial for face detections with the localization of facial landmarks and some detected bounding boxes by mistake are eliminated as shown in Fig. 6. Left-hand image is not through processing steps of facial landmarks. However, there are some wrong boxes, such as photo frame on the table, chair beside the window. On the contrary, right-hand image jointed detection and facial landmarks is shown significant results. In Table 3, we compare two results with F-net and without F-net. Obviously the results with F-net shows better performance on the datasets.

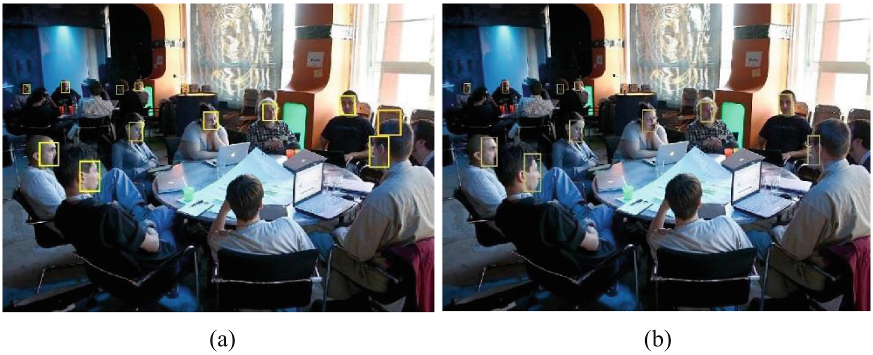


Fig. 6. Observing the result of facial landmarks. (a) The result of test image without facial landmarks. (b) The result of test image with facial landmarks.

Table 3. Comparison of our face detector with F-net and without F-net on datasets of WIDER FACE.

Method	Easy	Medium	Hard
With F-net	0.892	0.863	0.702
Without F-net	0.889	0.861	0.700

5 Conclusion

In this paper, we have proposed a novel cascaded CNNs framework for face detection and further gained additional performance by exploiting the inherent correlation between face detection and facial landmarks. Extensive evaluation on the challenging benchmarks for face detection demonstrate that our methods have achieved superior performance than the state-of-the-art methods. In the future, we will explore the correlation between super-resolution and facial features to further improve performance.

References

1. Farfade, S.S., Saberian, M.J., Li, L.J.: Multi-view face detection using deep convolutional neural networks. In: 5th ACM on International Conference on Multimedia Retrieval, pp. 643–650. ACM (2015)
2. Tran, C.K., Tseng, C.D., Lee, T.F.: Improving the face recognition accuracy under varying illumination conditions for local binary patterns and local ternary patterns based on weber-face and singular value decomposition. In: International Conference on Green Technology and Sustainable Development (GTSD), pp. 5–9. IEEE (2016)
3. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process. Lett. **23**(10), 1499–1503 (2016)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)

5. Lowe, D.G.: Object recognition from local scale-invariant features. In: Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
6. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**, 346–359 (2008)
7. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: Computer Vision and Pattern Recognition, CVPR, pp. 1–8 (2008)
8. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot multibox detector. In: arXiv preprint arXiv:1512.02325 (2015)
9. Roeder, N., Li, X.: Accuracy analysis for facial feature detection. Pattern Recogn. **29**, 143–157 (1996)
10. Yuille, A.L.: Deformable templates for face recognition. J. Cogn. Neurosci. **3**, 59–70 (1991)
11. Lam, K.M., Yan, H.: Locating and extracting the eye in human face images. Pattern Recogn. **29**(5), 771–779 (1996)
12. Deng, J.Y., Lai, F.: Region-based template deformation and masking for eye-feature extraction and description. Pattern Recogn. **30**, 403–419 (1997)
13. Renburgh, R.H., Clunies-Ross, C.W.: Linear discriminant analysis. Chicago **3**(6), 27–33 (1960)
14. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, p. 1. IEEE (2001)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Rabinovich, A: Going deeper with convolutions. In: Proceedings of the IEEE Conference On Computer Vision and Pattern Recognition, pp. 1–9 (2015)
17. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models-their training and application. Comput. Vis. Image Underst. **61**(1), 38–59 (1995)
18. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Trans. Pattern Anal. Mach. Intell. **23**(6), 681–685 (2001)
19. Cootes, T.F., Wheeler, G.V., Walker, K.N., et al.: View-based active appearance models. Image Vis. Comput. **20**(9), 657–664 (2002)
20. Yu, X., Huang, J., Zhang, S., Yan, W., Metaxas, D.N.: Pose-free facial landmark fitting via optimized part mixtures and cascaded deformable shape model. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1944–1951 (2013)
21. Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 157, pp. 2879–2886. IEEE Computer Society (2012)
22. Zhang, Z., Luo, P., Chen, C.L., Tang, X.: Facial Landmark detection by deep multi-task learning. In: European Conference on Computer Vision, vol. 8694, pp. 94–108 (2014)
23. Yang, B., Yan, J., Lei, Z., Li, S.Z.: Aggregate channel features for multi-view face detection. In: 2014 IEEE International Joint Conference on Biometrics (IJCB), pp. 1–8. IEEE (2014)
24. Yang, S., Luo, P., Chen, C.L., Tang, X.: Wider face: a face detection benchmark, pp. 5525–5533 (2015)
25. Yang, S., Luo, P., Loy, C.-C., Tang, X.: From facial parts responses to face detection: a deep learning approach. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3676–3684 (2015)
26. Ohn-Bar, E., Trivedi, M.M.: To boost or not to boost? On the limits of boosted trees for object detection. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 3350–3355. IEEE (2016)

27. Jia, Y., Shelhamer, E., Donahue, J., et al.: Caffe: convolutional architecture for fast feature embedding. In: In Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678. ACM (2014)
28. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
29. Seetaface Homepage. https://github.com/seetaface/SeetaFaceEngine. Accessed 2016

Cryptanalysis on the Anonymity of Li et al.'s Ciphertext-Policy Attribute-Based Encryption Scheme

Yi-Fan Tseng and Chun-I Fan(✉)

Department of Computer Science and Engineering,
National Sun Yat-sen University, Kaohsiung, Taiwan
yftseng1989@gmail.com, cifan@mail.cse.nsysu.edu.tw

Abstract. Attribute-based encryption is a very powerful primitive in public-key cryptography. It can be adopted in many applications, such as cloud storage, etc. To further protect the privacy of users, anonymity has been considered as an important property in an attribute-based encryption. In an anonymous attribute-based encryption, the access structure of a ciphertext is hidden from users. In this paper, we find an attack method against Li et al.'s anonymous attribute-based encryption schemes. The proposed attack uses an "invalid attribute key" to recover the hidden access structure of a given ciphertext. No information of the master secret key nor private keys are necessary in our attack.

Keywords: Attribute-based encryption · Anonymity · Hidden access structure Cryptanalysis

1 Introduction

As the Internet thriving over the whole world recently, there is a trend that everyone stores their sensitive data in third party storage space, such as Google drive or Dropbox. In order to protect the sensitive data from being revealed to attackers, public-key encryption is an appealing way to deal with the problem. In some scenarios, the identities of receivers may not be important, or the receivers are a group of users with same attributes, then this is where attribute-based encryption can be used.

Attribute-based encryption (ABE) is a branch in the researches of public-key cryptosystem, first introduced by Sahai and Waters [1] in 2005. In such encryptions, if the attributes satisfy some pre-defined access structures, then a ciphertext can be successfully decrypted. There are two types of ABE, key-policy attribute-based encryption (KP-ABE) [2–4] and ciphertext-policy attribute-based encryption (CP-ABE) [5–9]. The difference between these two kinds of ABE is where the attributes append on. In a KP-ABE scheme, an access policy will be assigned to a key when the key's holder enrolls into the system. A set of attributes will be appended to a ciphertext. On the other hand, in a CP-ABE scheme, a user's private key is associated with a set of attributes, and a ciphertext is assigned with an access structure.

In some applications for CP-ABE, the access structure of a ciphertext is necessary to be protected. Motivated by the requirement, a variant of CP-ABE called anonymous

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 98–104, 2018.
https://doi.org/10.1007/978-3-319-76451-1_10

CP-ABE or CP-ABE with hidden access structure is introduced [10–17]. In 2009, Li et al. proposed an anonymous CP-ABE [10]. The authors claimed that their scheme achieves short public parameters, provable security, and user accountability. However, we find an attack method to the anonymity of Li et al.'s scheme. Through our method, anyone can recover the access structure of a given ciphertext, even the user who has not registered in the system (i.e. the user who has not been issued a private key from the key generation center). Note that, in 2011 Li et al. [18] have mentioned a security flaw on the anonymity of [10], but we have proposed a much stronger result. In [18], Li et al. pointed out that the anonymity of [10] cannot be proven under a variant of DLIN assumption since the assumption is easy to solve through bilinear maps.

2 Preliminaries

2.1 Bilinear Maps

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic multiplicative groups of prime order p and g be a generator of $\mathbb{G}_1$. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a map with the following properties.

1. Bilinearity: $e\left(g_1^a, g_2^b\right) = e(g_1, g_2)^{ab}$ for all $g_1, g_2 \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p$.
2. Non-degeneracy: There exist $g_1, g_2 \in \mathbb{G}_1$ such that $e(g_1, g_2) \neq 1$.
3. Computability: There is an efficient algorithm to compute $e(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}_1$.

2.2 Access Structure

The access structure used in [10] is *AND-gate with multi-values*. Roughly speaking, each attribute owns multiple values. For example, the attribute could be CAREER, and its value could be Professor, Doctor, etc. Let n be the number of attributes in the system and $[1, n]$ be the integers from 1 to n. Each attribute ω_i contains a set S_i of values, and $|S_i| = n_i \in \mathbb{N}$, where $i \in [1, n]$. An access policy is defined as $W = [W_1, W_2, \ldots, W_n]$, where $W_i \subseteq S_i$ for $i \in [1, n]$. An attribute list $L = [L_1, L_2, \ldots, L_n]$ where $L_i \in S_i$ for $i \in [1, n]$ is said to satisfy an access structure $W = [W_1, W_2, \ldots, W_n]$ if $L_i \in W_i$ for $i \in [1, n]$.

3 The Review of Li et al.'s Scheme

There are three schemes proposed in [10]. In this section we briefly review the first scheme, anonymous CP-ABE with short public parameters. The scheme consists of four algorithms, **Setup**, **KeyGen**, **Enc**, **Dec**. The four algorithms are defined as follows.

- **Setup.** Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear map. Let the attribute universe $U = \{\omega_1, \omega_2, \ldots, \omega_n\}$ and S_i be the multi-value set for attribute ω_i, where $|S_i| = n_i$. The key generation center (KGC) performs as follows.

  1. Randomly choose $g_1, g_2 \in \mathbb{G}_1$ and $\alpha \in \mathbb{Z}_p$. Then compute $T = e(g_1, g_2)^{\alpha}$
  2. Choose a cryptographic hash function $H : \{0, 1\}^* \rightarrow \mathbb{G}_1$.

3. Publish the public parameter $para = (g_1, g_2, T, H)$, and keep secret the master secret key $msk = \alpha$.

- **KeyGen.** Given an attribute list $L = [L_1, L_2, \ldots, L_n]$, where $L_i \in S_i$. KGC performs as follows.

1. Pick randomly $s_1, s_2, \ldots, s_{n-1} \in \mathbb{Z}_p$ and compute $s_n = \alpha - \sum_{i=1}^{n-1} s_i$.
2. Choose randomly $r_i, r_i^{'} \in \mathbb{Z}_p, i \in [1, n]$.
3. For $i \in [1, n]$, compute the attribute key $sk_i = \{d_{i0}, d_{i1}, d_{i0}^{'}, d_{i1}^{'}\} = \{g_2^{s_i} H(1 \parallel i \parallel L_i)^{r_i}, g_1^{r_i}, g_1^{s_i} H(0 \parallel i \parallel L_i)^{r_i^{'}}, g_2^{r_i^{'}}\}$. The private key $sk_L = \{sk_1, sk_2, \ldots, sk_n\}$.

- **Enc.** To encrypt a message $M \in \mathbb{G}_2$ with an access structure $W = [W_1, W_2, \ldots, W_n]$, the encryptor performs as follows.

1. Choose randomly $z \in \mathbb{Z}_p$ and compute $C_0 = MT^z$.
2. For $i \in [1, n], t_i \in [1, n_i]$, if $v_{i,t_i} \in W_i$, choose $z_{i,t_i} \in \mathbb{Z}_p$ and compute $(C_{i,t_i,0}, C_{i,t_i,1}, C_{i,t_i,0}^{'}, C_{i,t_i,1}^{'}) = (H(1 \parallel i \parallel v_{i,t_i})^{z_{i,t_i}}, g_1^{z_{i,t_i}}, H(0 \parallel i \parallel v_{i,t_i})^{z - z_{i,t_i}}, g_2^{z - z_{i,t_i}})$; if $v_{i,t_i} \notin W_i$ choose $z_{i,t_i}, z_{i,t_i}^{'} \in \mathbb{Z}_p$ and compute $\left(C_{i,t_i,0}, C_{i,t_i,1}, C_{i,t_i,0}^{'}, C_{i,t_i,1}^{'}\right) = (H(1 \parallel i \parallel v_{i,t_i})^{z_{i,t_i}}, g_1^{z_{i,t_i}}, H(0 \parallel i \parallel v_{i,t_i})^{z_{i,t_i}^{'}}, g_2^{z_{i,t_i}^{'}})$.
3. The ciphertext is

$$C = \left(C_0, \left\{C_{i,t_i,0}, C_{i,t_i,1}, C_{i,t_i,0}^{'}, C_{i,t_i,1}^{'}\right\}_{i \in [1,n], t_i \in [1,n_i]}\right).$$

- **Dec.** Given a ciphertext $C = \left(C_0, \left\{C_{i,t_i,0}, C_{i,t_i,1}, C_{i,t_i,0}^{'}, C_{i,t_i,1}^{'}\right\}_{i \in [1,n], t_i \in [1,n_i]}\right)$, a user with private key $sk_L = \{d_{i0}, d_{i1}, d_{i0}^{'}, d_{i1}^{'}\}_{i \in [1,n]}$ on the attribute list $L = [v_{1,k_1}, v_{2,k_2}, \ldots, v_{n,k_n}]$ first compute

$$C' = \prod\nolimits_{i=1}^{n} \frac{e(C_{i,k_i,1}, d_{i0}) e\left(C_{i,k_i,1}^{'}, d_{i0}^{'}\right)}{e(C_{i,k_i,0}, d_{i1}) e\left(C_{i,k_i,0}^{'}, d_{i1}^{'}\right)},$$

and then compute $M = C_0 / C'$.

4 The Proposed Attack Method

In this section we will first give the high-level overview of our attack method, then give the detailed description of the proposed attack. Finally, we use a simple example to illustrate the scenario of the proposed attack.

4.1 Overview

The ciphertext C can be divided into three parts, the first part C_0 is the encryption of the message, which masks the message with the randomness z; the second part $\left(C_{i,t_i,0}, C_{i,t_i,1}\right)$ and the third part $\left(C'_{i,t_i,0}, C'_{i,t_i,1}\right)$ are used to help users with correct attributes to recover the information about z. The structures of second and the third parts are similar, though they are generated under different bases and randomness. We denote $\left(z_{i,t_i}, z'_{i,t_i}\right)$ as the ciphertext components $\left(C_{i,t_i,0}, C_{i,t_i,1}, C'_{i,t_i,0}, C'_{i,t_i,1}\right)$ such that $\left(C_{i,t_i,0}, C_{i,t_i,1}\right)$ and $\left(C'_{i,t_i,0}, C'_{i,t_i,1}\right)$ are generated under randomness $\left(z_{i,t_i}, z'_{i,t_i}\right)$ respectively.

Intuitively speaking, in the first scheme of [10], the way to achieve anonymity is to generate two types of ciphertexts, then argue that the two types are indistinguishable. If a value $v_{i,t_i} \in W_i$, then the ciphertext is $\left(z_{i,t_i} \in \mathbb{Z}_p, z'_{i,t_i} = z - z_{i,t_i}\right)$, which we call it "valid component"; otherwise, the ciphertext is $\left(z_{i,t_i} \in \mathbb{Z}_p, z'_{i,t_i} \in \mathbb{Z}_p\right)$. Using the ciphertext component $\left(z_{i,t_i}, z'_{i,t_i}\right)$ with corresponding attribute key, a user can obtain the term $e(g_1, g_2)^{s_i\left(z_{i,t_i} + z'_{i,t_i}\right)}$. In the case of "valid component", the user with correct attribute key obtains exactly $e(g_1, g_2)^{s_i z}$. Then she can compute $C' = \prod_{i=1}^{n} e(g_1, g_2)^{s_i z} = e(g_1, g_2)^{\alpha z}$, since $\sum_{i=1}^{n} s_i = \alpha$, and thus successfully decrypt the ciphertext.

The main concept of our attack is that we found that an attacker can generate an "invalid attribute key" without the information of *msk*. Although such a key cannot be used to recover the message, it can be used to distinguish whether a ciphertext component is "valid" or not. In our attack method, an attacker first computes an "invalid attribute" for attribute ω_i with value $k_i \in [1, n_i]$ under a randomness $\bar{s}_i$. If a ciphertext component $\left(z_{i,t_i}, z'_{i,t_i}\right)$ is a "valid component", then the attack should obtain $e(g_1, g_2)^{\bar{s}_i z}$. Since $\bar{s}_i$ is chosen by itself, the attacker can compute the term $e(g_1, g_2)^z$ if and only if the attribute ω_i with value k_i belongs to W_i, which means this relation can be used to distinguish the two types of ciphertexts.

4.2 The Proposed Attack

Given a ciphertext $C = \left(C_0, \left\{C_{i,t_i,0}, C_{i,t_i,1}, C'_{i,t_i,0}, C'_{i,t_i,1}\right\}_{i\in[1,n], t_i\in[1,n_i]}\right)$, an attacker performs the attack method as follows.

```
1. Set $W_1 = W_2 = \cdots = W_n = \phi$.
2. For $(i \in [1, n])$ {
     For $(t_i \in [1, n_i])$ {
       a. Choose $\bar{s_i}, r_i, r'_i \in \mathbb{Z}_p$.
       b. Compute "invalid attribute key" $(d_{i0}, d_{i1}, d'_{i0}, d'_{i1}) = \left(g_2^{\bar{s_i}} H(1 \| i \| v_{i,t_i})^{r_i}, g_1^{r_i}, g_1^{\bar{s_i}} H(0 \| i \| v_{i,t_i})^{r'_i}, g_2^{r'_i}\right)$
       c. Compute $x_{i,t_i} = \left(\frac{e(C_{i,k_i,1}, d_{i0}) e(C'_{i,k_i,1}, d'_{i0})}{e(C_{i,k_i,0}, d_{i1}) e(C'_{i,k_i,0}, d'_{i1})}\right)^{\frac{1}{\bar{s_i}}}$.
     }
     If there are two or more $x_{i,t_i}$ that are with same value, then add those $v_{i,t_i}$'s into $W_i$.
   }
Output the access structure $W = [W_1, W_2, \ldots, W_n]$.
```

4.3 Analysis

As we mentioned in Overview, the attacker generates an "invalid attribute key" for each value of each attribute, then uses it to decrypt the corresponding ciphertext. By the correctness of the Li et al.'s scheme, one can easily check that $x_{i,t_i} = e(g_1, g_2)^z$ iff v_{i,t_i} is chosen into the access structure, i.e. the ciphertext component is $(z_{i,t_i}, z - z_{i,t_i})$ for some $z_{i,t_i} \in \mathbb{Z}_p$. Moreover, if v_{i,t_i} is not in the access structure, which means the ciphertext component is $\left(z_{i,t_i} \in \mathbb{Z}_p, z'_{i,t_i} \in \mathbb{Z}_p\right)$, then $\Pr\left[z_{i,t_i} + z'_{i,t_i} = z\right]$ is negligible since z, z_{i,t_i}, z'_{i,t_i} are chosen uniformly at random from $\mathbb{Z}_p$. Though the attack algorithm seems necessarily to be performed over all the attributes in the universe, it is still practical. The reason is that the schemes in [10] only support small universe, i.e. the number of the total attributes is only polynomially many.

4.4 A Simple Illustration

In this section, we will give a simple illustration to demonstrate the proposed attack. Assume that there is a small system that only contains one attribute with three values $\{A, B, C\}$. Therefore, the universe is $\{\omega_1 = \{A, B, C\}\}$, and the public parameter *para* and the master secret key *msk* is the same as the setting in Sect. 3. Assume that the attacker is given a ciphertext C encrypted under the access structure $W = [W_1 = \{A, C\}]$, i.e.

$$C = \left(\begin{array}{cc} & \left\{ C_{1,1,0}, C_{1,1,1}, C'_{1,1,0}, C'_{1,1,1} \right\}, \\ C_0, & \left\{ C_{1,2,0}, C_{1,2,1}, C'_{1,2,0}, C'_{1,2,1} \right\}, \\ & \left\{ C_{1,3,0}, C_{1,3,1}, C'_{1,3,0}, C'_{1,3,1} \right\} \end{array} \right)$$

$$= \left(\begin{array}{cc} & \left\{ H(1 \parallel 1 \parallel A)^{z_{1,1}}, g_1^{z_{1,1}}, H(0 \parallel 1 \parallel A)^{z-z_{1,1}}, g_2^{z-z_{1,1}} \right\}, \\ MT^z, & \left\{ H(1 \parallel 1 \parallel B)^{z_{1,2}}, g_1^{z_{1,2}}, H(0 \parallel 1 \parallel B)^{z'_{1,2}}, g_2^{z'_{1,2}} \right\}, \\ & \left\{ H(1 \parallel 1 \parallel C)^{z_{1,3}}, g_1^{z_{1,3}}, H(0 \parallel 1 \parallel C)^{z-z_{1,3}}, g_2^{z-z_{1,3}} \right\} \end{array} \right)$$

To recover the hidden access structure, the attacker first generates "invalid attribute key" for value, i.e.

$$sk_A = \left(g_2^{\bar{s}_1} H(1 \parallel 1 \parallel A)^{r_1}, g_1^{r_1}, g_1^{\bar{s}_1} H(0 \parallel 1 \parallel A)^{r'_1}, g_2^{r'_1} \right),$$

and computes $x_{1,1} = \left(e(g_1, g_2)^{\bar{s}_1 z} \right)^{\frac{1}{\bar{s}_1}} = e(g_1, g_2)^z$. Similarly, the attacker can obtain $x_{1,2} = e(g_1, g_2)^{z_{1,2} + z'_{1,2}}, x_{1,3} = e(g_1, g_2)^z$. Since $x_{1,1} = x_{1,3}$, the access structure $W = [\{A, C\}]$ is recovered, and thus the anonymity is violated.

5 Conclusion

After reviewing Li et al.'s paper [10], we found that their schemes do not achieve anonymity. Anyone can produce "invalid attribute keys" to recover the hidden access structure of a ciphertext. The proposed attack does not need any information of the master secret key nor private keys. Though we only break the anonymity of the first scheme of [10], the other two schemes in [10] have very similar construction to the first one, and actually they are designed based on the first scheme. Therefore, we believe there are the same flaws in the other two schemes in [10].

Acknowledgement. This work was partially supported by the Ministry of Science and Technology of the Taiwan under grants MOST 105-2923-E-110-001-MY3, MOST 105-2221-E-110-053-MY2, and MOST 106-3114-E-110-001.

References

1. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Advances in Cryptology – Eurocrypt, vol. 3494. LNCS, pp. 457–473. Springer (2005)
2. Attrapadung, N., Libert, B., de Panafieu, E.: Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: International Workshop on Public Key Cryptography, pp. 90–108. Springer (2011)

3. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, pp. 89–98 (2006)
4. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 195–203. ACM (2007)
5. Bethencournt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy, SP 2007, pp. 321–334 (2007)
6. Cheung, L., Newprot, C.: Provably secure ciphertext policy ABE. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, pp. 456–465 ACM (2007)
7. Goyal, V., Jain, A., Pandey, O., Sahai, A: Bounded ciphertext policy attribute based encryption. In: Automata, Languages and Programing, pp. 579–591. Springer (2008)
8. Liang, X., Cao, Z., Lin, H., Xing, D.: Provably secure and efficient bounded ciphertext-policy attribute-based encryption. In: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security, pp. 343–352. ACM (2009)
9. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Public Key Cryptography. Lecture Notes in Computer Science, pp. 53–70 (2011)
10. Li, J., Ren, K., Zhu, B., Wan, Z.: Privacy-aware attribute-based encryption with user accountability. In: International Conference on Information Security, vol. 9, pp. 347–362 Springer (2009)
11. Balu, A., Kuppusamy, K.: Ciphertext-policy attribute-based encryption with anonymous access policy. arXiv preprint arXiv:1011.0527 (2010)
12. Balu, A., Kuppusamy, K.: Privacy preserving ciphertext-policy attribute-based encryption. In: Recent Trends in Network Security and Applications, pp. 402–409. Springer (2010)
13. Lai, J., Deng, R.H, Li, Y.: Fully secure cipertext-policy hiding CP-ABE. In: Information Security Practice and Experience, pp. 24–39. Springer (2011)
14. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Applied Cryptography and Network Security, pp. 111–129. Springer (2008)
15. Padhya, M., Jinwala, D.: A novel approach for searchable CP-ABE with hidden ciphertext-policy. In: Information Systems Security, pp. 167–184. Springer (2014)
16. Phuong, T.V.X., Yang, G., Susilo, W.: Hidden ciphertext policy attribute-based encryption under standard assumptions. IEEE Trans. Inf. Forensics Secur. **11**(1), 35–45 (2016)
17. Wang, Z., He, M.: CP-ABE with hidden policy from waters efficient construction. Int. J. Distrib. Sens. Netw. **2016**, 11 (2016)
18. Li, J., et al.: Multi-authority ciphertext-policy attribute-based encryption with accountability. In: Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security. ACM (2011)

Overlapping Community Detection with Two-Level Expansion by Local Clustering Coefficients

Yi-Jen Su(✉) and Che-Chun Lee

Shu-Te University, Yanchao District, Kaohsiung City 82445, Taiwan
iansu@stu.edu.tw

Abstract. Community detection is crucial to Social Network Analysis (SNA) in that it helps to discover high-density overlapping communities hidden in complex networks for advanced applications. This study proposed a novel community detection method by seed set expansion. The method gathered meaningful nodes into a seed set, which was then used as a central node to merge neighbor nodes until communities were found. To enhance efficiency, a two-level expansion approach was further developed, which adopted the 80/20 rule and involved threshold change in order to discover cohesive subgroups of smaller sizes. To detect overlapping communities, local clustering coefficients (LCC) were calculated to measure the interaction density between neighbor nodes and determine whether they expanded or not. The experiment results were evaluated by measuring the cohesion quality of communities.

Keywords: Social network analysis · Community detection
Clustering coefficients

1 Introduction

In social network analysis, depending on the number of groups that a node belongs to, community detection methods can be divided into two categories: partitioning-based methods and overlapping-based methods. In partitioning-based methods, an actor/node belongs to one group only. The most famous method in this category is the GN (Girvan-Newman) algorithm [1], which is a top-down method that repeatedly removes the highest-betweenness edge from the network to incrementally find groups. On the other hand, overlapping-based methods move bottom-up, with an actor/node being present in several groups. For example, the CPM (Clique Percolation Method) algorithm [2] expands k-cliques to detect overlapping communities. In previous studies, the quality of community detection was often measured by the modularity, the value of quality Q [3], or cluster cohesion [4].

Aiming to reduce computation time by avoiding the need for a full scan, some local-based methods have been proposed, for example, seed set expansion. The seed set expansion method is a local link-based analysis that was first presented in the HITS algorithm [5] proposed by Jon Kleinberg in 1999. The technique has since been widely adopted in community finding [6] and webpage community discovery [7].

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 105–112, 2018.
https://doi.org/10.1007/978-3-319-76451-1_11

Overall, seed set expansion methods can be divided into three main groups, including k-means, high-degree, and random walk. The k-means methods adopt Graclus [8] to split data into k clusters, using their centroids as seeds. Most seeds are presented by link-based methods as the centroids of cohesive subgroups. The high-degree methods follow the hypothesis that a high-degree node and its neighbor nodes can merge into a cohesive group with a high clustering coefficients. The easiest way to find candidate seeds is to randomly select nodes as seed set members from a given graph, thereby saving computation time.

Seed set expansion methods are similar to clustering methods, especially those with seed nodes in k-means or nodes with higher degrees. The community detection process adopts the discovered seeds as the basis to incrementally grow by adding new member nodes in the neighborhood to the original seed set until reaching a cohesive group. The process of seed set expansion terminates when no more nodes are qualified to be joined to the growing group. In previous research, however, the network was decomposed to find communities. Seed set expansion was first completed by using random-walk to find new member nodes [9]. Then the conductance distance of neighbor nodes was computed for low conductance cuts [10].

According to the concept of 20/80 distribution, 80% of connections are linked to 20% of nodes. Applying the same idea to overlapping community detection, this study adopted degree centrality and used nodes with higher degree in the graph/network as seed set members. This seed set generation method effectively reduced the execution time as it simply followed the descending order of nodes' degree until reaching 80% degree of the whole graph. In the seed set expansion process, seed nodes were retrieved one by one, in descending degree order, to calculate the local clustering coefficients (LCC) [11] of all neighbor nodes to decide whether to add a neighbor node to the group or not. Both the execution time and the subgroup quality were used as criteria to check the effectiveness of the proposed method, as compared with the CPM, a well-known overlapping community detection method.

2 Literature Review

2.1 Quality of Community Detection

In 1906, Italian economist Vilfredo Pareto first proposed the 80/20 rule [12] when he observed that 20% of the Italian population owned 80% of property in the country. Based on the 80/20 rule, this study assumed that only a few nodes own a large number of neighbor nodes, while most nodes connect to only a few nodes in complex networks. Using the top 20% of nodes as seeds can help to detect most communities in complex networks.

Community detection is an important research issue in SNA. The most cohesive subgroup in a complex network is a complete subgraph, also called a "clique", where all member node pairs are connected by an edge. In large-scale complex networks, a high-quality result of community detection holds a much higher intra-relationship density of communities than the inter-relationship density between communities [13].

The most popular method to evaluate detection result is Clustering Coefficients (CC). The CC of community C_i is calculated as Eq. (1):

$$CC_i = \frac{\left|e_{jk}:v_j, v_k \in N_i, e_{jk} \in E_i\right|}{\left(k_i * \left(k_i - 1\right)\right)/2} \tag{1}$$

where community C_i contains a set of nodes N_i and a set of edges E_i. e_{jk} presents the relationship between a pair of nodes v_j and v_k belonging to N_i. k_i stands for the number of nodes in community C_i.

On the other hand, the Local Clustering Coefficients (LCC) is adopted to measure the relationship density between neighbors of a node, for example the LCC of node v_i as Eq. (2).

$$LCC\left(v_i\right) = \frac{\left|e_{jk}:v_j, v_k \in N_{v_i}, e_{jk} \in E_{v_i}\right|}{\left(d_i * \left(d_i - 1\right)\right)/2} \tag{2}$$

where e_{jk} shows an edge existing between a pair of nodes, v_j and v_k, and both nodes are neighbor nodes of node vi. In addition, N_{vi} is the set of all neighbor nodes of node v_i and di is the degree of node vi. E_{vi} is the set of all edges between neighbor nodes of node vi.

In 2004, M. E. J. Newman and M. Girvan first proposed the concept Modularity, the value of quality Q, for measuring the overall subgrouping quality resulting from partition-based community detection. In overlapping community detection, a node may belong to several communities simultaneously. When Graph G=(V,E) includes M edges, the computation of modularity Q can be retrieved as shown in Eq. (3).

$$Q = \frac{1}{2M} \sum_{(i,j \in VxV)} \left[A_{ij} - \frac{k_i k_j}{2M}\right] \frac{1}{O_i O_j} \tag{3}$$

where o_i and o_j represent vertices v_i and v_j belonging to a number of communities. k_i and k_j are the degrees of vertices v_i and v_j, respectively. An adjacent matrix A is adopted to represent the relations between all nodes in Graph G=(V,E) with M edges. The content of each cell A_{ij} is either 1 or 0.

2.2 Seed Set Expansion

Seed set expansion, first proposed by Jon Kleinberg in hyperlink analysis for webpage ranking algorithm HITS [5], has been wildly used in network community discovery and webpage subgrouping. In recent years, seed set expansion has been applied to SNA-related research, such as overlapping community detection.

All seed set members are selected and collected from important nodes of the network structure. The collection methods variously depend on the degree of nodes, the total distance to all other member nodes, or random selection. The purpose of forming seed sets is to collect more important nodes for high-quality grouping. Gleich [9, 14] maintained that seed set collection is more important than seed set expansion in community detection.

3 Research Method

This research proposed a novel overlapping community method, of which the operation sequence divided into three phases, as shown in Fig. 1. The first phase was Data Preprocessing, which recursively removed all noise nodes, whose degree was either 0 or 1, from the complex network until no more noise nodes existed. Next, in the Seeding phase, all nodes of the pruned network were ranked in the descending order of the nodes' degree. Based on the 80/20 rule, only the top 20% nodes were checked with Local Clustering Coefficients (LCC) whether greater than the threshold H_{lcc} or not. If the LCC of a node was higher than the threshold H_{lcc}, the node would be added to the seed set. Last, in the Seed Set Expansion Phase, the seed nodes were selected sequentially as the center of a new community and incrementally added its community neighbors to the current community when the LCC was higher than the threshold H_{lcc}. In order to avoid skipping smaller-scale communities, the Expansion phase adopted another higher LCC threshold H_{lcc} to check if the remaining nodes formed small-scale communities. Once a remaining node could serve as a new seed, the seed set expansion process would be executed again.

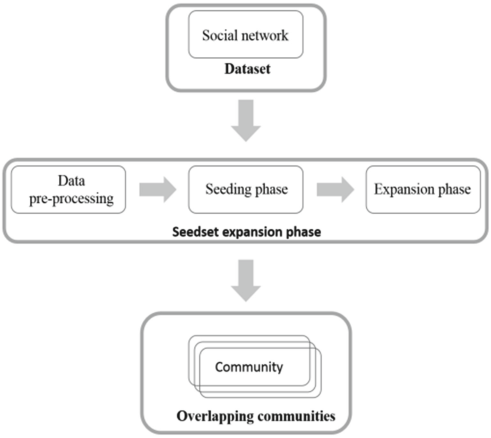


Fig. 1. The operation of overlapping community detection by LCC

3.1 Seed Set Construction

The seeding method of this research modified Gleich's approach [14] by inserting higher-degree nodes into the seed set. It followed the 80/20 rule and adopted the top 20% of nodes, which presumably owned 80% of the connections in the complex network, to construct the seed set. The details of the operation are shown in Table 1.

Table 1. Seeding two algorithm.

```
Seeding Phase
Input : graph G = (V, E) , graph G' = (V, E)
Output : seedset S
Threshold H = |V| * 20%;
seedset S = ∅;
While |S| <= H do
    Find node $v_i$ with max degree in G'
    $S = S \cup v_i$;
    G' = G' - edges($v_i$);
Return S
```

3.2 Seed Set Expansion

First, following the order of nodes' degree from the highest to lowest, a node was retrieved from the seed set to function as the center of a community. Next, this study used local clustering coefficients (LCC) to decide whether all neighbor nodes of the center should be merged into the current community, as shown in Table 2. Once the LCC value was higher than the threshold T_{lcc}, which means that there is strong cohesion between the center and its neighbors, then the community would expand while neighbor nodes were added. In expansion operations like depth-first search, all member nodes' neighbors would be checked by LCC threshold T_{lcc} recursively. The expansion operation would stop when there was no neighbor node to satisfy the T_{lcc} constraint.

Table 2. Community expansion algorithm.

```
Growth function
Function Growth($v_i$, $c_i$, Threshold $T_{lcc}$, markN)
    If markN.contains($v_i$) != ture
        markN = markN $\cup$ $v_i$;
        If LCC($v_i$) $\geq T_{lcc}$ do
            $c_i = c_i \cup$ neighbor($v_i$);
            For $v_j \in$ neighbor($v_i$) do
                $c_i = c_i \cup$ Growth($v_j$, $c_i$, $T_{lcc}$);
Return $c_i$
```

In expansion level one, all members of the seed set were checked with LCC computation iteratively, as shown in Table 3. When the LCC of a seeding node exceeded the threshold, initially the seeding node would serve as the center of a community and incrementally added neighbors into the community recursively by LCC checking. However, with an LCC lower than the threshold, the seeding nodes were skipped. All checked nodes were marked and would not be checked in expansion level two.

Table 3. Seed set expansion level one algorithm.

```
Expansion Level-1 Phase
Input : graph $G=(V,E)$; seedset S; Threshold $T_1$ ; Community(C) $c_i(i = 0,\ldots,k-1)$; markN
Output : Communities
For $v_i \in S$ do
    Initialize $c_i = \varnothing$;
    $C = C \cup$ Growth($v_i$, $c_i$, $T_1$, markN);
Return C
```

In expansion level two, the checking targets were nodes without checking marks. As the purpose of this operation was small-scale high-cohesion community detection, the LCC threshold was adjusted higher than expansion level one, as shown in Table 4.

Table 4. Seed set expansion level-two algorithm.

```
Expansion Level-2 Phase
Input : graph $G=(V,E)$; Threshold $T_2 > T_1$; Community(C) $c_i(i = 0,\ldots,k-1)$; markN
Output : Communities
For $v_i \in V$
    Initialize $c_i = \varnothing$;
    If $v_i \notin$ markN do
        $C = C \cup$ Growth($v_i$, $c_i$, $T_2$, markN);
Return C
```

4 Experiment Results

The experiment adopted five datasets retrieved from SNAP (Stanford Network Analysis Project) and Network Repository, including three co-authored networks CA-GrQc, CA-CondMat, CA-HepPh and two Facebook datasets, as shown in Table 5. The input networks that came from Facebook owned higher-density relationship between members, but the relation density of all co-authored networks was sparser.

Table 5. Five datasets and detailed information.

Dataset	No. of nodes	No. of edges	Density
CA-GrQc	5,242	14,484	0.00105
CA-CondMat	23,133	93,439	0.00034
CA-HepPh	12,008	118,489	0.00164
Facebook-1	4,039	88,234	0.01081
Facebook-2	1,446	59,589	0.057

The experiment result showed that the CPM constructed a smaller number of communities with many member nodes as shown in Table 6. The proposed LCC method generated a larger number of communities with few member nodes. The communities constructed by the CPM were looser; therefore, the method might work better with sparse complex networks, for example, the dataset Ca-CondMat. On the other hand, the proposed LCC method, stricter in seed expansion, might produce better result when applied to complex networks with higher-density relationships, for example, the dataset Facebook-1.

Table 6. The Comparison of CPM and the Proposed LLC Method.

Dataset	No. of communities		Node-ratio in communities		Clustering coefficient		Modularity Q	
	CPM	LCC	CPM	LCC	CPM	LCC	CPM	LCC
CA-GrQc	831	1037	73%	72%	0.92	0.92	0.55	0.56
Facebook-1	16	107	98%	98%	0.76	0.86	0.19	0.37
CA-CondMat	2897	4537	88%	87%	0.92	0.90	0.62	0.44

As the proposed LCC method reduced execution time in several steps, it was consistently faster than the CPM with all the five datasets, as shown in Fig. 2. Notably, once the scale of the input network became larger, for example, CA-HepPh, though the clustering coefficients of the two methods remained the same at 0.93, the LCC method spent only 15% of CPM's execution time.

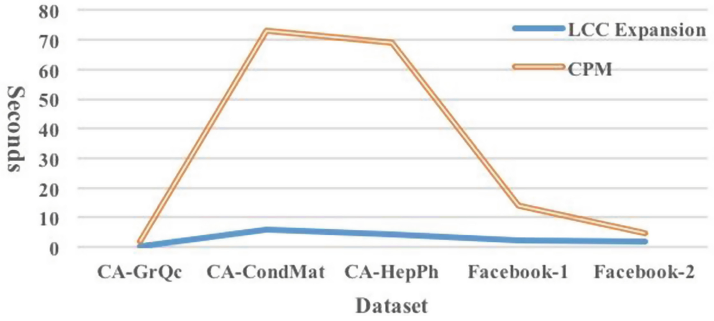


Fig. 2. Comparison of execution time: CPM vs the proposed LCC method.

5 Conclusion

The research proposed a LCC-based expansion method to detect overlapping communities in complex networks. The method not only yielded high-quality detection results but also significantly reduced execution time. The proposed community detection method was divided into three phases: data pre-processing, seed set construction, and seed set expansion. First, in data pre-processing, low interaction nodes, whose degree was 0 or 1, were viewed as noise from the input graph and recursively removed. Next, in the seed set construction phase, the top 20%-degree nodes were gathered to form the seed net. Last, in the expansion phase, with each qualified seed node, whose LCC was higher than the threshold T_{lcc}, serving as the center of a new community, all qualified neighbors were incrementally added to the community. However, the three-phase operation only revealed larger scale of communities, with smaller-size communities easily being skipped. Therefore, the remaining nodes were all checked by a higher LCC threshold to detect smaller-scale cohesion subgroups.

References

1. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Phys. Rev. E **69**(6), 066133 (2004)
2. Derényi, I., Palla, G., Vicsek, T.: Clique percolation in random networks. Phys. Rev. Lett. **94**(16), 160202 (2005)
3. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
4. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, pp. 29–42 (2007)
5. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
6. Flake, G., Lawrence, S., Lee Giles, C.: Efficient identification of web communities. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 150–160 (2000)
7. Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A.: Trawling the Web for emerging cyber-communities. Comput. Netw. **31**(11–16), 1481–1493 (1999)
8. Dhillon, I., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors: a multilevel approach. IEEE Trans. Pattern Anal. Mach. Intell. **29**(11), 1944–1957 (2007)
9. Whang, J.J., Gleich, D.F., Dhillon, I.S.: Overlapping community detection using seed set expansion. In: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, pp. 2099–2108 (2013)
10. Havemann, F., Heinz, M., Struck, A., Glaser, J.: Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. J. Stat. Mech. Theor. Exp. **2011**, P01023 (2011)
11. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1988)
12. Furlan, V.: Vilfredo Pareto, Manuale di Economia Politica. Jahrbücher für Nationalökonomie und Statistik **91**(1), 826–831 (1908)
13. Turner, J.C.: Towards a cognitive redefinition of thesocial group. Social identity and intergroup, pp. 15–40 (1982)
14. Whang, J., Gleich, D., Dhillon, I.: Overlapping community detection using neighborhood-inflated seed expansion. IEEE Trans. Knowl. Data Eng. **28**(5), 1272–1284 (2016)

IoT and E-commerce Applications

Writing Security Specification with Things That Flow

Sabah Al-Fedaghi[1(✉)] and Omar Alsumait[2]

[1] Computer Engineering Department, Kuwait University, P.O. Box 5969, 13060 Safat, Kuwait
sabah.alfedaghi@ku.edu.kw

[2] Information Technology Department, Ministry of Defense, Safat, Kuwait
oaalsumait@mod.gov.kw

Abstract. In the field of security, writing a Request For Proposals (RFP) includes a description of specifications that requires careful definition of problems and an overview of how the system works. An important aspect in this context is how to generate technical specifications within the RFP. This "specification writing" is a complex subject that causes even design professionals such as architects and engineers to struggle. Typically an RFP is described in English, with graphs and tables, resulting in imprecise specifications of requirements. It has been proposed that conceptual representation such as UML diagrams and BPMN notations be included in any RFP. This paper examines RFP development of Public Key Infrastructure (PKI) and proposes a conceptual depiction as a supplement to the RFP to clarify requirements more precisely than traditional tools such as natural language, tables, and ad hoc graphs. A case study of an actual government ministry is presented with a model, i.e., diagrams that express how the features and services of PKI would logically operate in the requisite system.

Keywords: RFP · Public key infrastructure · Conceptual modeling
Diagrammatic representation

1 Introduction

A Request For Proposals (RFP) is normally a basic tool used to solicit submission of proposals from prospective vendors for a desired service. Security administrators usually issue an RFP when purchasing security-related items such as a public key infrastructure (PKI) system or an intrusion alarm system. Writing an RFP requires carefully defining the problems that need solving and explaining how the system to be purchased works. An important question in this context is how to generate technical specifications within the RFP. This "specification writing" "is a complex subject that even design professionals such as architects and engineers struggle with" [14]. There are many specification methods, including [14],

- Proprietary Method: the buyer provides a detailed product description of what is to be bought and how to install it.
- Performance Method: the buyer describes the desired end result, but leaves details of accomplishing this result to the bidder.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 115–125, 2018.
https://doi.org/10.1007/978-3-319-76451-1_12

Most organizations have some form of standard terms and conditions that they typically attach to RFPs and other contract documents. Usually, the same terms and conditions are used to purchase anything from a box of paperclips to a tractor-trailer rig and they contain many *"requirements that may not be applicable to the typical security or surveillance project"* [14] (Italics added).

In general, it is quite common to see RFPs with requirements that are very broad, derived from a vendor's list of features, or copied from another organization's RFP [9]. According to Hadrian and Evequoz [12], producing more precise requirement specifications would be helpful for all stakeholders. Requirements should be unambiguous and validated by business users. The authors enumerate the main difficulties in RFP requirements specification, including expressing precisely what will be needed (i.e., specific requirements) and attaching requirements to specific parts in a process.

Businesses seeking software solutions are advised to "Model your business process graphically. Business process diagrams (or models) are excellent at showing gaps in the process or errors in your understanding" [9]; this source particularly recommends Swim Lane diagrams. Many works have been published that utilize diagrams for modeling a procurement process such as RFP at the operational level. Douraid et al. [10] modeled the static and dynamic behavior of a procurement system in UML that includes an ordering process. The Electoral Officer of Canada [11] used BPMN in an RFP for automation of the polling process. According to Electoral Officer of Canada [11], over the past few decades, the polling process in Canadian General Elections has become increasingly complex. In response it was decided to introduce electronic devices that automate paperwork and other administrative tasks performed by election officials in the polling place. The RFP is envisioned as a *conceptual* E-Poll Solution, with requirements structured around that vision. State diagrams (Fig. 1) are utilized to clarify the requirements, and the narrative is supplemented with business process model notation (BPMN) diagrams (Fig. 2).

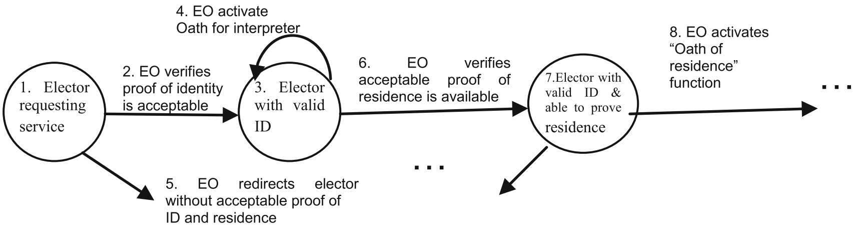

4. EO activate Oath for interpreter
1. Elector requesting service
2. EO verifies proof of identity is acceptable
3. Elector with valid ID
6. EO verifies acceptable proof of residence is available
7.Elector with valid ID & able to prove residence
8. EO activates "Oath of residence" function
5. EO redirects elector without acceptable proof of ID and residence
...
...

Fig. 1. State diagram for electors (redrawn, partial from [11])

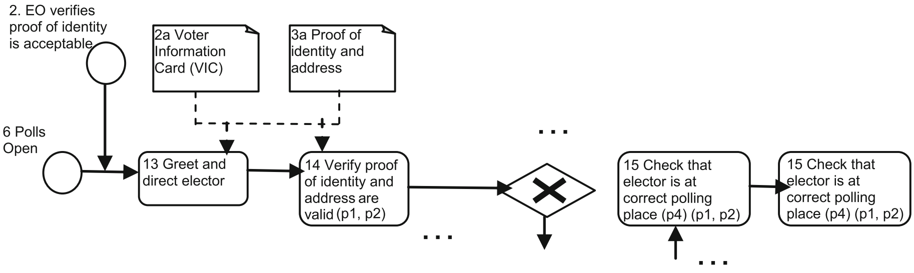

2. EO verifies proof of identity is acceptable
2a Voter Information Card (VIC)
3a Proof of identity and address
6 Polls Open
13 Greet and direct elector
14 Verify proof of identity and address are valid (p1, p2)
15 Check that elector is at correct polling place (p4) (p1, p2)
15 Check that elector is at correct polling place (p4) (p1, p2)
...
...
...

Fig. 2. BPMN check-in (redrawn, partial from [11])

Similarly, this paper proposes a conceptual model, the Flowthing Machine (FM) model that can be used to facilitate the creation of RFP specifications. The resulting model can then be understood by all stakeholders without particular knowledge of technical details. The diagrams in FM express the technical parts of the RFP in a "neutral" representation that facilitates communication among stakeholders.

We selected PKI as the content of a demonstration RFP because "all of the books or Web sites on the subject either assume that you already know all about PKI or they use so many big words that they are hard for a beginner to understand" [13]. PKI is suitable as a test case for communication among stakeholders by providing a nontechnical language that underlies the RFP.

For the sake of a self-contained paper, the next section briefly reviews the FM model that forms the foundation of the theoretical development in this paper. The FM diagrammatic language has been adapted to several applications [2–8]; however, the example given here is a new contribution.

2 Flowthing Model

FM uses "flowthings" (hereafter, *things*) to model a range of items – for example, ordered items, certificates, keys signals, employees, signatures, data, and so on – and their dynamic behavior. *Things* flow in (abstract) *machines* among basic stages in which a thing can be created, released, transferred, processed, and received (see Fig. 3).

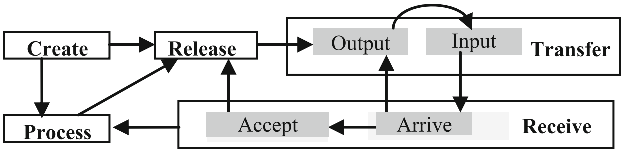


Fig. 3. Flowthing machine

The machine is the conceptual structure used to change or transmit things as they pass through stages, from their inception or arrival to their de-creation or transmission. Machines form the organizational structure (blueprint) of any system. These machines can be embedded in a network of assemblies called *spheres* (e.g., Encryption system, Certificate Authority System) in which the machines operate.

The stages in Fig. 3 can be described as follows:

Arrive: A thing reaches a new machine.
Accept: A thing is permitted to enter, or not. If arriving things are always accepted, Arrive and Accept can be combined as a *Receive* stage.
Process (change): A thing goes through some kind of transformation that changes it without creating a new thing.
Release: A thing is marked as ready to be transferred outside the machine.
Transfer: A thing is transported somewhere to or from outside the machine.
Create: A new thing appears in a machine.

The machine shown in Fig. 3 is a generalization of the typical input-process-output (IPO; see Fig. 4), a process model used in many scientific fields. According to Aagesen and Krogstie [1], most process modeling languages take an approach built on IPO:

Input
Process
Output

Fig. 4. Input-process-output model

Processes are divided into activities, which may be divided further into subactivities. Each activity takes inputs, which it transforms to outputs. Input and output relations thus define the sequence of work. This perspective is chosen for the standards of the Workflow Management Coalition (WfMC 2000), the Internet Engineering Task Force (IETF), the Object Management Group (OMG 2000), IDEF (1993), Data Flow Diagram, Activity Diagrams, Event-driven Process Chains, BPMN and Petri nets.

The stages in FM are mutually exclusive. An additional stage of Storage can also be added to any machine to represent the storage of things; however, storage is not an exclusive stage, because there can be stored processed flowthings, stored created flowthings, etc. In FM, the notion of spheres and subspheres refers to network environments. Multiple machines can exist in a sphere if needed. The machine is a subsphere that embodies the flow; it itself has no subspheres.

Additionally, in FM, triggering, denoted by a dashed arrow, indicates the activation of a flow. It is a dependency among flows and parts of flows. A flow is said to be triggered if it is created or activated by another flow. Triggering can also be used to initiate events such as starting up a machine.

Example. Talhi et al. [15] investigated the main approaches adopted for specifying and enforcing security in UML design.

Since there is neither consensus nor standard on how security should be specified for UML design, non-security experts designers are feeling lost when it comes to deal with security aspects of their design. In fact they are looking for precise answers to many questions. Unfortunately, as far as we know, the state of the art is not providing such precise answers. In fact we did not find any contribution covering all these aspects in the same study providing UML designers with the expected answers [15].

Talhi et al. [15] discuss UML approaches, including activity diagrams, to show how security requirements can be specified for UML design. Figure 5 specifies system behavior in relation to the admission of patients to a medical institution and consists of three main partitions: (1) Patient, who starts the activity by filling out an admission request, (2) Administration area where insurance and cost information are collected, and (3) Medical area responsible for admission tests, exams, medical evaluations, and sending the medical results to the patient.

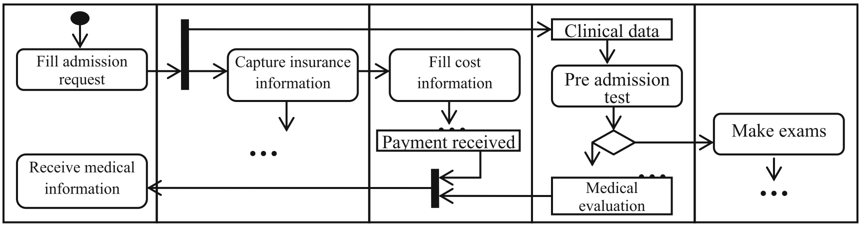


Fig. 5. Enforcing security requirements in an activity diagram (redrawn, partial from [15]).

Here, we encourage comparison of the activity diagram with a corresponding FM representation in order to gain a general appreciation of FM diagrams (see Fig. 6).

In Fig. 6, the patient creates an Admission request (circle 1) that flows to Admission (2). For simplicity, note that boxes around some subspheres, e.g., Admission request, have been omitted. Also note that arrows are drawn in different colors to emphasize different stages in the flow of the request. Upon receipt of the request in Admission, it is processed and,

- The insurance information is extracted (4) and
- The request is sent to Medical evaluation (5)

Upon receipt of the request in Medical evaluation (6), it is processed (7) to trigger the generation of a pre-admission test (8) that triggers the performance of exams (9) that are processed (10) to trigger the creation of clinical information (11), that flows to Medical Evaluation (12), then to Accounting (13).

After the insurance information (4) is extracted, it flows to Accounting for calculation of cost (14). This triggers creating a request to the patient to pay (15, 16). Upon receiving this invoice, the patient creates a payment (17) and sends it to the Accounting department (18). Receipt of the payment (18) releases the clinical information (19) that is sent to the patient (20).

In UML, the activity diagram is considered a representation of behavior. In FM, Fig. 6 is considered a static script that would be “eventized”: cut into pieces according to the natural joints of possible events, to represent behavior. The resulting time-based schemata are used to *control and manage* execution of the system. An event is a *thing* that can be created, processed, received, released, and transferred in time. Time is also a *thing* that can also be created, processed, received, released, and transferred. Note the uniform conceptualization of things that flow, including events and time. No additional notation is needed to model the system behavior.

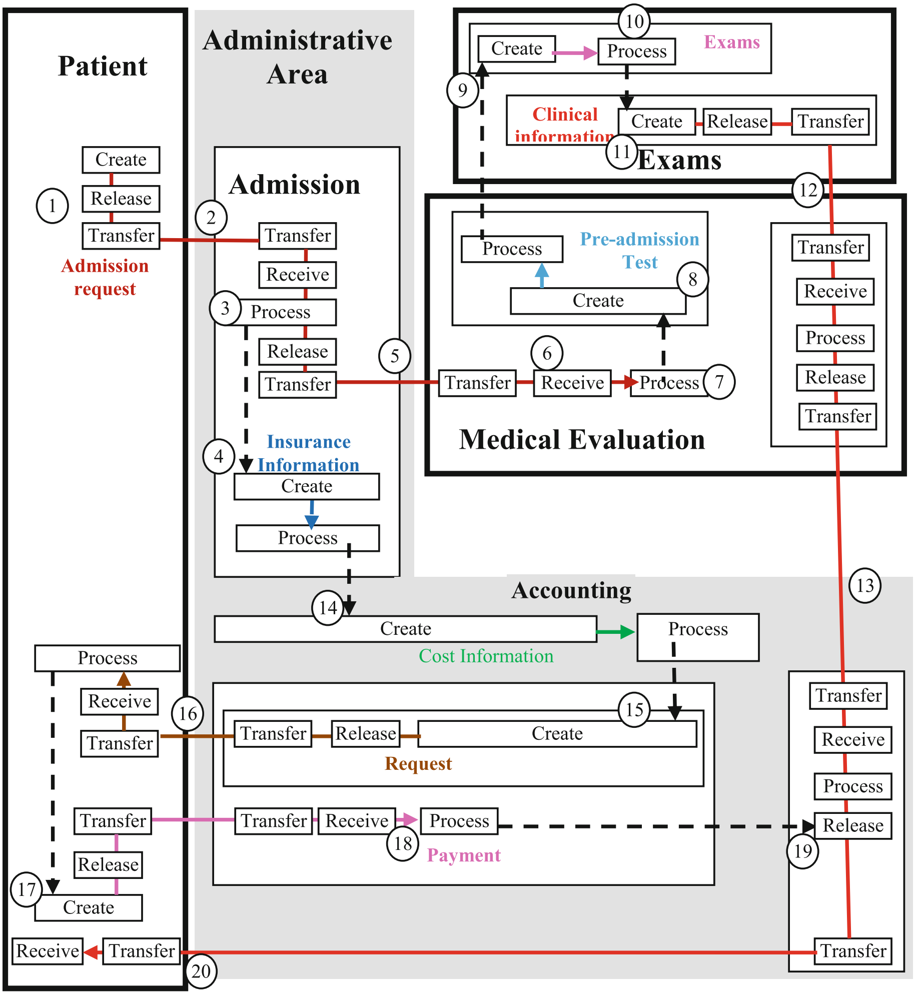


Fig. 6. FM representation corresponding to the activity diagram of Fig. 5

The sphere of an *event* in FM comprises the machines of time and the event itself. Accordingly, Fig. 7 shows the event ***A patient generates an admission request that flows to Admission in the Administrative area***. In the figure the *Process* of an event (top flow in Fig. 7) indicates an event running its course. For simplicity, we will omit the stages of the event and its time machine from the representation of events.

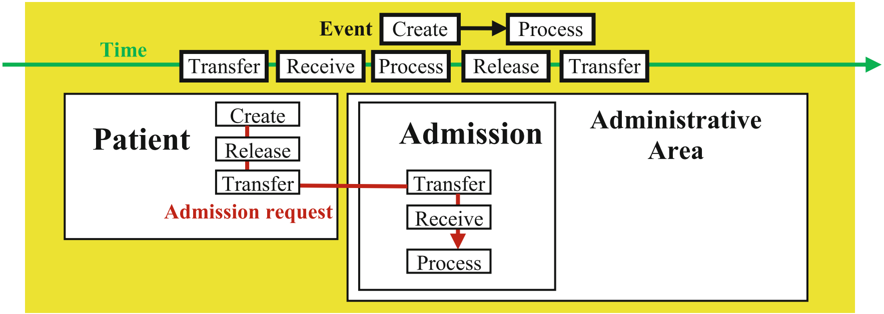


Fig. 7. The event: *A patient generates an admission request that flows to Admission in the Administrative area*

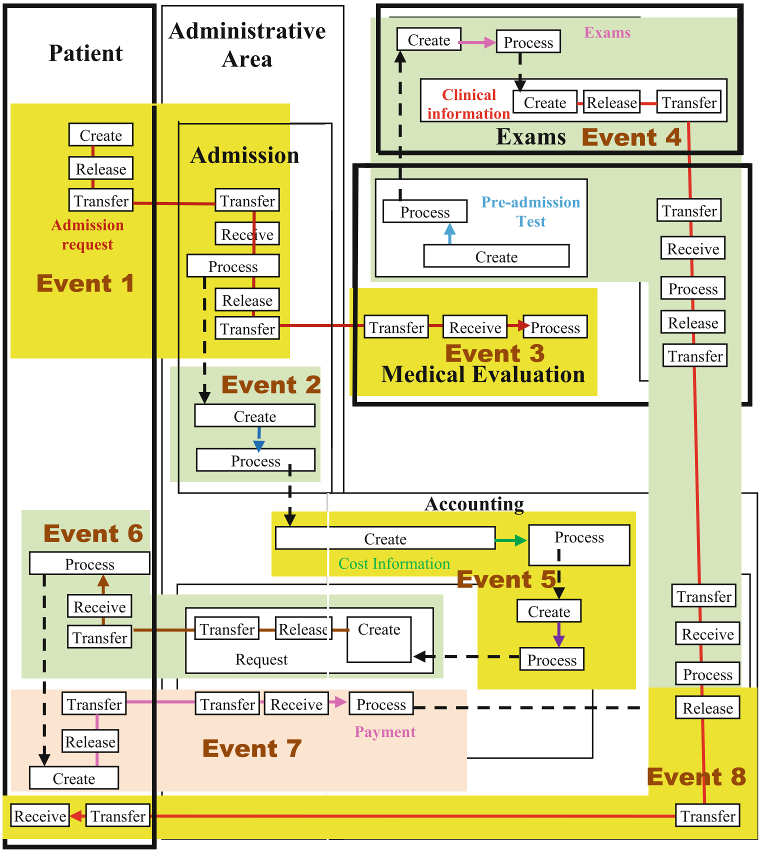


Fig. 8. FM representation of events

Figure 8 shows selected “meaningful” events in the example:

Event 1: *A patient generates an admission request that flows to Admission in the Administrative area*
Event 2: *Admission extracts insurance information*
Event 3: *Admission sends a request for medical evaluation*
Event 4: *Exams are conducted*
Event 5: *Accounting calculates cost*
Event 6: *Accounting requests payment from patient*
Event 7: *Patient makes payment*
Event 8: *Clinical information is sent to patient*

Now the total picture of the FM-based system is complete. The functional and behavioral components have already been described. This description is followed by developing *control* in the form of mechanisms to guide or regulate system behavior so it functions as intended.

3 Case Study

This section reports the results of applying FM to develop an RFP for a government ministry seeking Enterprise PKI service that facilitates introducing digital signatures to authenticate the identity of the sender of a message or signer of a document. The ministry has decided to provide digital signature capabilities as a framework for performing (internal) electronic services in a secure manner that ensures the integrity of transactions.

The PKI provides authentication and a digital signature for government employees. Here we assume general knowledge of public key cryptography; i.e., a digital signature requires a key pair: the Public and Private Keys. Note that each model in this section represents a cyber-physical system that integrates computation with physical processes, where users interact utilizing both their digital identities and the physical interaction.

Even though the main objective of the project is to identify and implement the most appropriate PKI solution that fulfills the ministry's *requirements for improving the security of its IT Infrastructure,* it is unclear what these requirements are. We will focus here on the parts that describe digital signatures. This part of the system includes a conceptual model that helps in describing how the required system registers users and issues PKI certificates, and additional description (not included) contains how the system is used by the employees of the ministry.

Figure 9 shows the FM representation of the ministry conceptualization of the issuance of digital signature certificates under the PKI framework. In the figure, an employee (circle 1) applies for a digital certificate via his/her account (2). The request flows (3) to the web interface that sends (4) it to the CryptoServer (server dedicated to Biometric-PKI). This triggers (5) the release of the applicant's fingerprint. The received fingerprint (6) triggers the creation of public and private keys (7 and 8) used in the Biometric-PKI system. Accordingly, the Registration Authority (RA) receives (9) the created keys and validates the applicant's identity, then passes them to the Certification Authority (CA) (10). The CA (11) combines the public key (13) with the validated data (12) to create a digital key certificate (14).

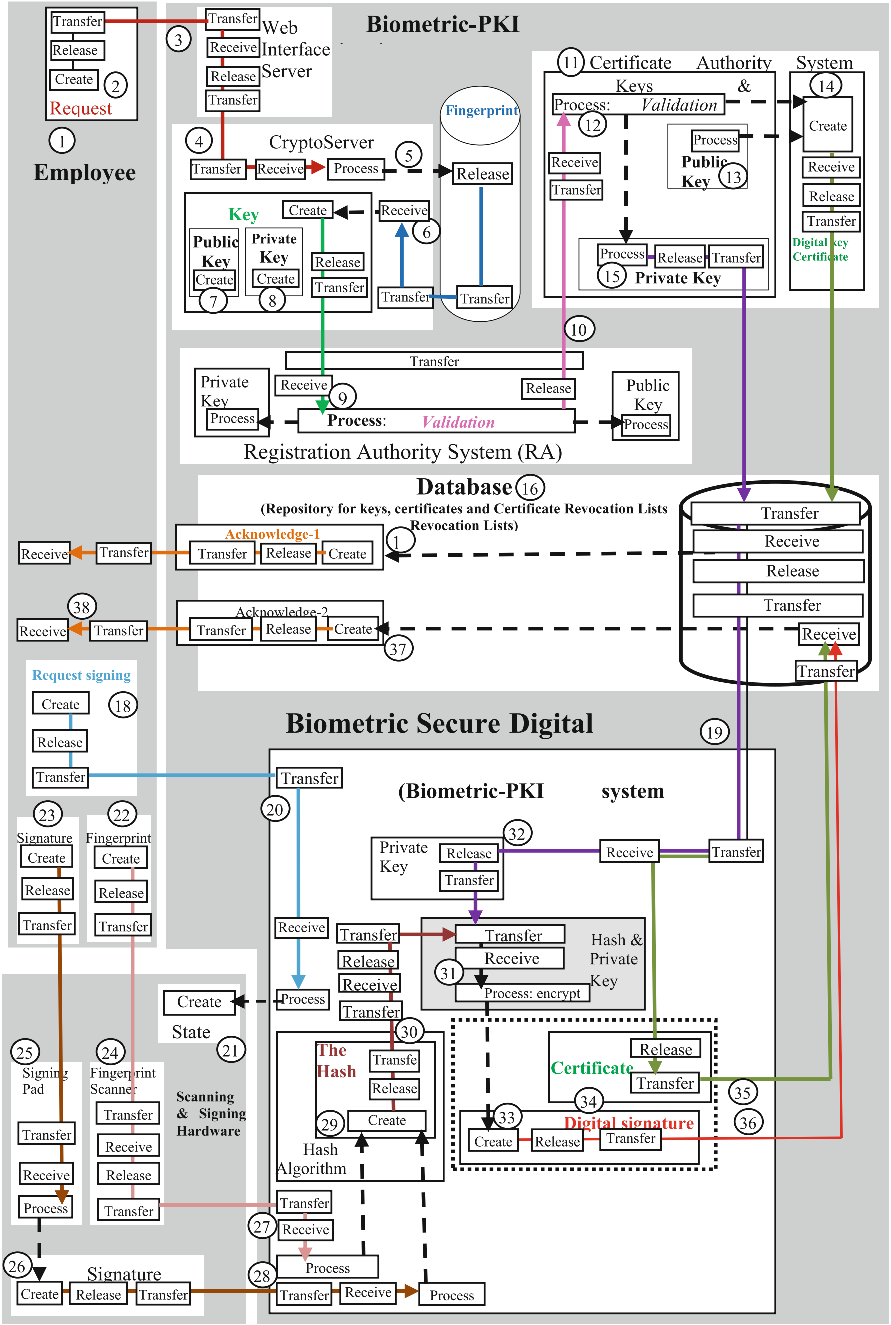


Fig. 9. FM description of the digital certificate process as conceptualized by the Ministry.

The private key (15) and the digital key certificate are sent and stored in the database (16). This triggers the database to issue an acknowledgment-1 (17) which instructs the user that the system is ready for the next step (18): signing. The private key and the digital key certificate are forwarded to the Biometric Secure Digital Signature system (19). The Employee activates the Scanning & Signing hardware (20 and 21) to provide his/her fingerprint (22) and signature (23) to the Scanning & Signing hardware (24, 25). The user enters his/her signature to trigger the creation of a digital signature image (26). The Biometric-PKI system server receives both the signature image and the fingerprint (27 and 28), to trigger creation of the Hash (29) using the Hash Algorithm. The created Hash flows (30) to be combined with the private key (31) sent earlier from the repository (32) to create a digital signature (33) to be combined with the certificate (34). The digital signature and the certificate flow together (35, 36) to the Database (Repository) to trigger the creation of Acknowledgment-2 (37) that flows to the employee (38).

Figure 9 is supposed to be attached to the RFP of the PKI and discussed with the various bidders by stakeholders in the ministry. Thus a rigid method is not imposed; rather it is an initial "vision" of the problem that the agency tries to solve; and the bidder can respond with a counter model that is a modification or replacement of this FM conceptualization. Of course different modifications are expected during these discussions.

The representation can also be used for other purposes during the project, including identification of acceptance tests.

4 Conclusions

The paper has proposed utilizing a diagrammatic conceptual representation known as FM as a tool for the specification of requirements in RFPs. FM is applied to a sample case study of RFP for public key infrastructure (PKI). The results indicate that FM is viable as a modeling tool that complements RFP. FM diagrams may present difficulties because of their seeming complexity; however, some solutions to this problem have already been implemented in engineering systems through multilevel simplifications. Further studies will investigate other types of RFPs.

References

1. Aagesen, G., Krogstie, J.: Analysis and design of business processes using BPMN. In: vom Brocke, J., Rosemann, M. (eds.) Handbook on Business Process Management 1, International Handbooks on Information Systems. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-00416-2_10
2. Al-Fedaghi, S.: Business process modeling: blueprinting. Int. J. Comput. Sci. Inf. Secur. **15**(3), 286–291 (2017)
3. Al-Fedaghi, S.: Flow-based process modeling: application in BPMN and process-oriented software systems. In: Cybernetics Approaches in Intelligent Systems, pp. 86–98. Springer (2018). https://doi.org/10.1007/978-3-319-67618-0_9
4. Al-Fedaghi, S.: Conceptual modeling in simulation: a representation that assimilates events. Int. J. Adv. Comput. Sci. Appl. **7**(10), 281–289 (2016)

5. Al-Fedaghi, S.: Design functional decomposition based on flow. In: IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2016), Budapest, 9–12 October 2016
6. Al-Fedaghi, S.: Diagrammatic modeling language for conceptual design of technical systems: a way to achieve creativity. Int. Rev. Autom. Control **9**(4) (2016)
7. Al-Fedaghi, S., Alahmad, H.: Integrated modeling methodologies and languages. In: ACM 12th International Conference on Ubiquitous Information Management and Communication, Langkawi, 5–7 January 2018
8. Al-Fedaghi, S., Alahmad, H.: Orientation in conceptual modeling frameworks. In: The 3rd IEEE International Conference on Big Data Intelligence and Computing, Orlando, 6–10 November 2017
9. F.H. Black & Company: How to Prepare Better RFP Requirements Lists for IT Success, CaseWare (2017). https://www.caseware.com/us/2017/04/27/prepare-better-rfp-requirements-lists-success?lang=es
10. Douraid, A., Elhaq, S.L., Ech-Cheikh, H.: A conceptual and UML models of procurement process for simulation framework. Int. J. Comput. Sci. Issues (IJCSI) **9**(6, no. 1) (2012)
11. Electoral Officer of Canada: Request for Proposal, Voting Services Modernization/Polling Place Process Enhancement, File No. ECRS-RFP-16-0167, 22 June 2017. https://buyandsell.gc.ca/cds/public/2017/06/23/734d6a2085fb71d89ff3c02b95a0c2cf/ecrs-rfp-16-0167_voting_services_modernization-polling_place_process_enhancement.pdf
12. Hadrian, D., Evequoz, F.: CARES: Requirements Specification with BPMN 2.0 in WTO Procurement. Institut d'Informatique de Gestion, HES-SO Valais-Wallis (2014). http://publications.hevs.ch/index.php/attachments/single/974
13. Posey, B.: A Beginner's Guide to Public Key Infrastructure: PKI Can Help Keep Your Network Secure, But It Can Be a Hard Concept to Understand, 15 September 2005. http://www.techrepublic.com/article/a-beginners-guide-to-public-key-infrastructure/
14. Silva Consultants: Writing an Effective RFP for Security Systems (2017). http://www.silvaconsultants.com/writing-an-effective-rfp-for-security-systems.html
15. Talhi, C., Mouheb, D., Lima, V., Debbabi, M., Wang, L., Pourzandi, M.: Usability of security specification approaches for UML design: a survey. J. Object Technol. **8**(6), 103–122 (2009). http://www.jot.fm/issues/issue_2009_09/article1/

Automatically Generating Aspect Taxonomy for E-Commerce Domains to Assist Sentiment Mining

Nachiappan Chockalingam(✉)

Department of Computer Science and Engineering,
College of Engineering, Guindy, Chennai 600025, Tamil Nadu, India
nach729@hotmail.com

Abstract. Numerous reviews are available online for many domains, and increasingly even for singular products. In this scenario, aspect associations to domains can be made extensive. Instead of generating aspects from the training set of reviews for a domain, the task of aspect generation is pushed onto an automated taxonomy generation system. Based on certain user input parameters, the taxonomy is expanded using an unsupervised web crawl of E-Commerce Website(s). The aspect taxonomy can be used to assist researchers in annotation of reviews to use for training classifiers for sentiment analysis, and for visualization of sentiment analysis results.

Keywords: Sentiment mining · Dataset · Aspect
Dependency parsing · Taxonomy

1 Introduction

This paper outlines a solution to create a taxonomy consisting of aspects of an E-Commerce domain. Taxonomies have been built for numerous domains (e.g.: agriculture [1], medicine [5]). They have been used for knowledge representation and for building more "intelligent" applications. In the context of sentiment analysis, an aspect taxonomy can be used for annotation assistance and sentiment analysis representation.

In the sentiment mining field, there are numerous product review datasets available for researchers to use. But with fast evolving fields like E-Commerce, analysis algorithms (Artificial Neural Network, CNN, etc.) as well as higher computational power, review datasets need to be both large and current. A widely used dataset from Hu's paper [7] has reviews of a 'DVD Player' and 'MP3 Player' extracted in 2004. How long has it been since those you've heard those terms? Additionally, there exist only a couple hundred reviews for each Domain. Other data sets including those from Ding et al. [3] and Popescu et al. [11] exhibit similar problems.

They do justice neither to product evolution nor help exploit better computational power to work with larger datasets that newer algorithms need for

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 126–137, 2018.
https://doi.org/10.1007/978-3-319-76451-1_13

training (explained in application section). The idea expressed in this paper assists in dealing with the pitfalls mentioned above by decreasing time taken for review annotation. Additionally, visualization of sentiment analysis results are often presented in a confusing manner with polarity scores for aspects given individually. If the domain is organized in a clear hierarchy, the analysis results of customer perception of the product is easier to grasp. Eg: in the mobilephone domain-

mobilephone
- **camera**
- - lens
is a better representation than,
mobilephone
- **camera**
- **lens**

1.1 Application

The generated taxonomy will have the sentences associated with an aspect linked to it. For example, the aspect "Cost" will have all the 200 sentences associated with its identification as an aspect, linked to it. Since E-Commerce Domain Aspect taxonomies are uncommon, some applications are illustrated below.

1. We see an trend in the industry where sentiment models need more test data. [12] Rosenthal et al.'s 2017 semEval challenge uses CrowdFlower to get both the sentiment and feature annotation while a group of Graduate Students were tasked with the same in Zimbra et al.'s 2016 paper [17] which uses an Artificial Neural Network. Thousands of datasets are annotated at great time and expense. With each broad domain (e.g.: Movie, E-Commerce, etc.) requiring its own train and test datasets - lots of productive time is spent on classifying reviews. By applying the proposed system, the annotator need only assign sentiment scores for the review set, and perform manual aspect classification of sentences that did not get classified under a taxonomy aspect. Thereby saving time on classifying the entire dataset. A possible concern with the use of Crowd Platforms for annotation is, it might not yield good accuracy as annotators might not be experienced in classification or simply disinterested in the work.
2. Use the generated taxonomy to help visualize the final sentiment mining results, similar to [16]. Sentiment scores assigned to the aspects in a taxonomy can be easily understood by an analyst. Efforts are hence put in to ensure human readability of the taxonomy, so that the it can be modified (insertion or deletion) and understood easily.

Overall, this model can scale well for practical use in Aspect Based Sentiment Analysis. Any erroneous aspect classifications can be easily spotted after the taxonomy is built, and hence removed. The corresponding sentences can be pushed to the unclassified set. Likewise, aspects can be added to prepare the taxonomy for sentiment analysis visualization.

2 Related Work

Yu et al.'s paper [16] is work that attempts to do something similar to the idea expressed in this paper. The hierarchy based on lexico-syntactic relations between aspects are expressed using Hearst Patterns. However the end goal expressed by them is only to create an organization that would allow the user to easily grasp the overview of the reviews. Their system while sentiment neutral, uses a sentiment analysis based approach to identify aspects (Pros and Cons [15]).

On the other hand, Garcia et al. [6] attempted to create an aspect list for Restaurant and Laptop (E-Commerce) domains from a semEval training set by extracting aspects from them. They further made an effort to classify identified aspects into categories.

Identification of aspects has been done [9] using supervised, unsupervised and semi-supervised methods. Our goal is to automatically create a taxonomy in an iterative manner, and hence we use an unsupervised aspect identification method.

2.1 Novelty

While Yu et al. [16] do reference their hierarchy for aspect identification, they never formally presented it for building intelligent systems and there has been very little research on automated learning of aspects using automatic crawling. Citing these factors, the technique presented in this paper stands apart in opening up the aspect annotation and sentiment analysis presentation to a new approach.

3 Proposed System

The central idea behind this system is, a successful aspect identification of a relevant aspect will not need further review, with the assumption that a single sentence contains reference to only one aspect. For example, If a sentence "I liked the case a lot" is classified under aspect "case" - the chance that aspect classification is wrong is small. But say, case is irrelevant to our domain, the aspect is discarded from the taxonomy and the sentences associated with it are sent to the unclassified section.

The system can be split into 3 broad modules, the Review Crawler, Aspects Generator and Taxonomy Insertion.

3.1 Review Crawler

The Review Crawler fetches reviews from E-Commerce Websites based on the input parameters. In the paper [4] "Ontology Focused Crawling of Web Documents", the authors emphasize the advantage of "focused crawling". "Focus" is implemented by limiting the search to E-Commerce Websites. Instead of using search engines or link-extraction (from web pages), an E-Commerce Website's

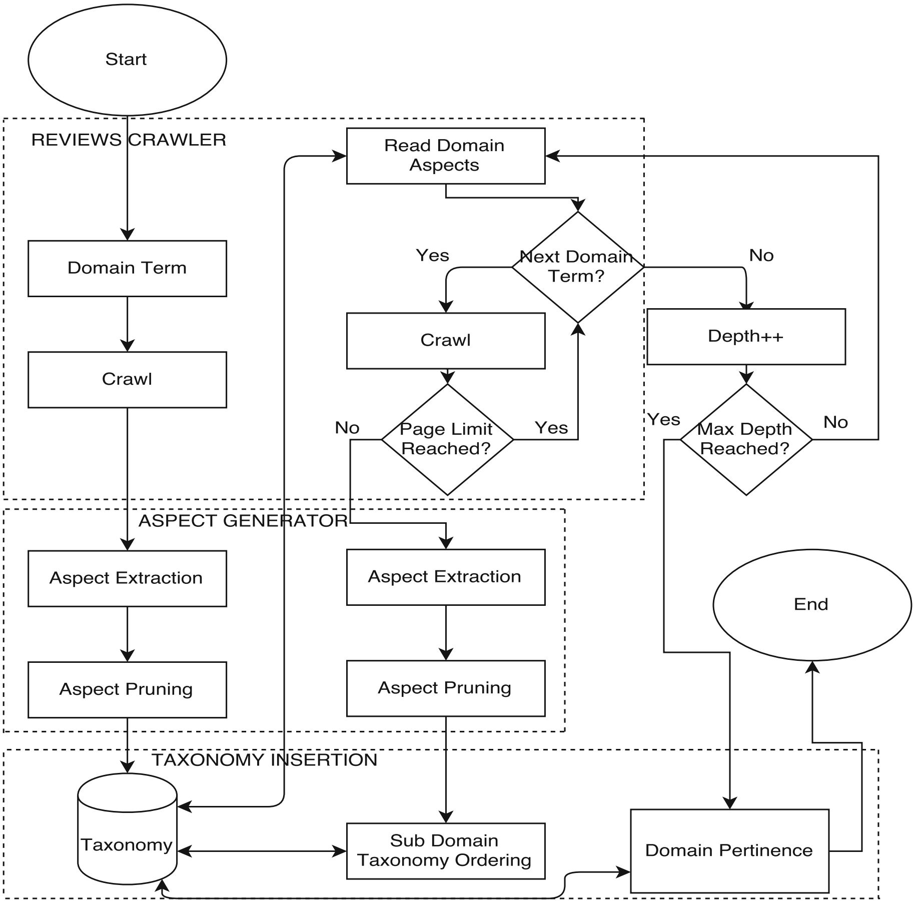


Fig. 1. Proposed system diagram

search functionality is used to obtain products of the domain, whose reviews are then crawled. It could be construed that the E-Commerce website is used as a search engine to get links to product reviews (Fig. 1).

After aspect extraction, and insertion of master domain's terms into the taxonomy, the aspects are taken as domain terms and crawled. This is done in an iterative manner until some threshold is met.

Spam and Duplicates Handling: Duplicates are removed. Reviews with a bag of words cosine similarity higher than 0.6 with each other are discarded.

3.2 Aspect Generator

The Aspect Generator takes in reviews of a domain term and outputs the generated aspects of the domain.

Cleaning Data: The reviews are tokenized into sentences (the analysis is done on the sentence level). Punctuation marks are removed next and so are non-ASCII characters and all characters are made lower case. Lastly, conjunctions ("but", "and", etc.) are handled by either splitting the sentence into two if they are standalone (compound sentences) sentences or leaving the sentence as it is if the 2 sentences cannot exist independently. We need to ensure 1 sentence has only 1 aspect associated with it to the greatest degree possible (since that is an assumption we make for this system).

Analysis: Mukherjee et al. [10] present a method using dependency parsing relations (dbj, nmod, etc.) between the words of a sentence, to determine the opinion word and the aspects of a domain. An effective unsupervised technique to obtain aspects. We cannot use even semi-supervised techniques such as double propagation (using aspect seed words) because the crawl is supposed to be iterative and details are unavailable about the domains that will be crawled, except for the master domain term the taxonomy is being created for. The extracted features are unigram at this stage. Eg: The dependency parser relationship nmod is relevant and connects 'camera' with 'great'. Hence, 'camera' is extracted as an aspect (Fig. 2).

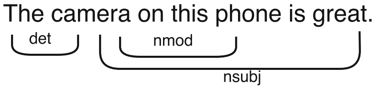


Fig. 2. Unsupervised aspect identification

Pruning Aspect Set: Once aspect words have been generated, noise needs to be reduced. The frequent-frequency technique [7,15] is used to limit the number of aspects. Another useful method is eliminating aspects with sentiment scores. Senti-Word Net sentiment scores is used for this purpose - if an aspect has a non 0 sentiment score, it is discarded. Stop-Word removal and lemmatization are also done to prune the aspect set. Lemmatization is preferred over Porter Stemmer because it maintains human readability which is necessary for our application objective. Example:

Computers would become:
Porter Stemmer: Comput
Lemmatizer: Computer

Detecting Multi-words: Efforts to detect Multi-Words, like performing a posteriori pruning for adjacent nouns, and taking the most frequent ones, leads to a lot of noise. Using domain neutral WordNet [6] to detect compound phrase entries perform poorly because the domains are different.

Compound phrases like "battery life" can be captured, instead, by using dependency parsing. If "battery" and "life" are both features, and are linked by the compound dependency parser tag, they are merged as one feature - "battery life".

3.3 Taxonomy Model

Taxonomy insertion deals with inserting the extracted Aspects into the taxonomy as its elements. We need to determine a cutoff page value for our review crawling for an aspect. Ideally, this formula should incorporate the relationship between the domain to be crawled and the Master Domain (Domain Term), and begin limiting the pages as it proceeds to the next level of the taxonomy (aspect occurrence in master domain's value). A formula to accomplish this is proposed.

Formula: The maximum page size for crawling of a term is expressed as:

$$pageSize = \frac{masterReviewSize * log(aspectOccurenceInMasterDomain)}{(k * reviewsPerPage)} \tag{1}$$

where k is a constant determined by the user and masterReviewSize is the size of the master term's review set size. If the crawler is unable to get a pre-determined number of reviews within the page size, or if the crawler runs out of reviews to crawl (even if still within page size), the aspect crawling is aborted.

Domain Pertinence: There may be duplicate aspect identifications. But an ideal aspect taxonomy should have an aspect appear only once. To achieve this goal, Domain Pertinence [2] is applied. If the value is greater than 1, the domain is classified under term 'i', otherwise under term 'j' in the taxonomy. The formula for domain pertinence is given below:

$$Pertinence = \frac{freq(t/Di)}{(freq(t/Dj))} * \frac{sentencesSize(Dj)}{sentencesSize(Di)} \tag{2}$$

Where freq(t/Di) represents the frequency of a term in Domain 'i', and freq(t/Dj) represents the frequency of the term in Domain 'j'. The comparison needs to be scaled since the size of sentence list in each domain differs after data cleaning. Hence the review size of domain 'j' is divided by review size of domain 'i'.

To illustrate how it functions: the word "cable" is identified as an aspect of "mobilephone" (master domain term) and "charger". When the aspect of "charger" is to be inserted into the taxonomy, the above formula is applied. If the result shows it to be more pertinent to the cable domain, it is removed from the "mobilephone" aspect list and placed under "charger" along with aspect terms (if it has any). By applying it repeatedly for duplicates, no aspect term will appear more than once in the system. Aspect identification data about how many times an aspect has been mentioned in the reviews needs to have been cached for this stage.

A deadlock will be reached when two domains have the other as an aspect. Eg when "camera" has lens as an aspect, and "lens" has "camera" as an aspect. The aspect with higher affinity to the master domain in distance first or frequency second is given precedence. Another issue that arises is problem of often used terms like “cost” classified under some random term. In this case, a threshold for classification is set - if an aspect is identified in more than a certain number of domains - it is removed from the domain pertinence pool and assigned to the highest term it appears under or simply discarded.

4 Evaluation

4.1 Experiment

The crawler was coded for Amazon and the evaluation was carried out for the mobilephone domain. It is a rich domain with numerous aspects and is one of the most popular of E-Commerce products. The generated taxonomy is expected to have a reasonable number of features with 6000 reviews being used to identify them.

The aspect generator was built atop NLTK and Stanford Dependency Parser packages on Python. It takes in the initial 6000 reviews and generates 50 aspects which is put into the taxonomy under the master term ‘mobilephone’.

Next, each identified aspect is again put through the crawl system and the aspect identification takes place in an iterative manner. The top 20 extracted aspects are inserted into the taxonomy under the term whose aspect they were identified as. Each identified aspect also has the set of reviews it is mentioned in, linked to it.

4.2 Observations

1. As the review set gets larger, the change in aspect ranking (based on frequency) reduced. In other words adding more reviews does not improve/affect aspect identification.
2. Regarding the aspect set, certain words like “product” were seen to occur frequently in every domain. Domain pertinence elimination where a threshold set to remove aspects in too many fields was effective.
3. Modifications can be applied to some aspects while crawling for their aspects. Features like “battery life” fails to pass the review extraction page limit. In this case splitting the words and running the E-Commerce domain search with the more frequently used term (battery was used more frequently than life) helped. Seemingly arbitrary, but proves to be an effective solution (Fig. 3).
4. Prior to Domain Pertinence application, if a domain term’s features have already been identified, the aspects can be placed under it without engaging in crawling or analysis. For example, “screen” is classified under “mobilephone” and “camera”, but has already been crawled for “mobilephone”. The aspect set of “screen” can be placed under “camera” without replicating the crawl and analysis.

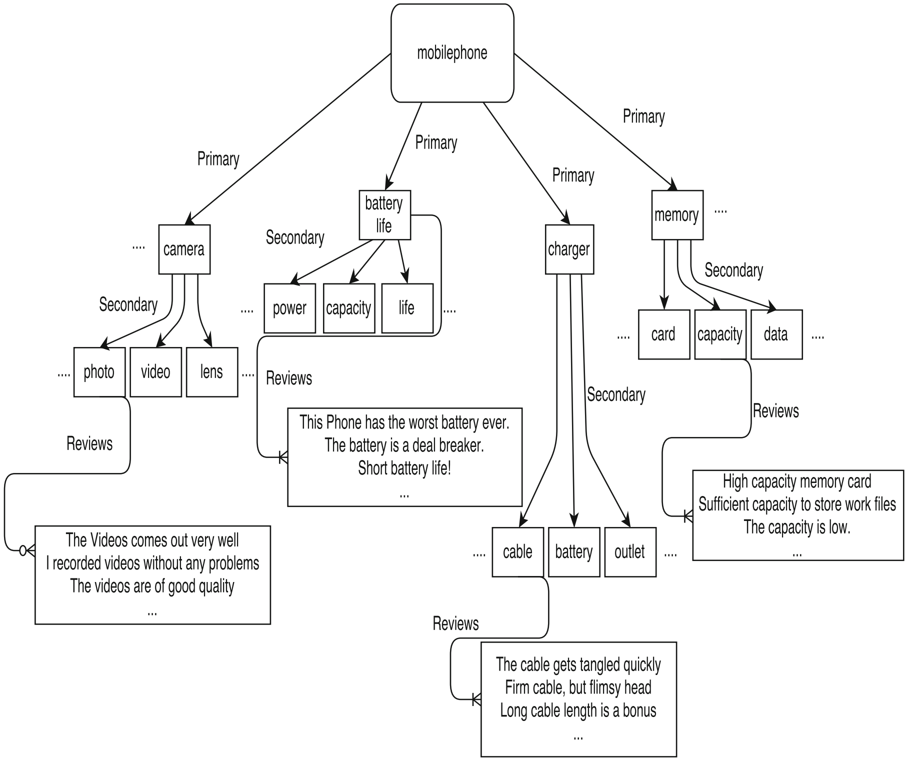

mobilephone
Primary
Primary
Primary
Primary
camera
battery life
charger
memory
Secondary
Secondary
Secondary
Secondary
photo
video
lens
power
capacity
life
cable
battery
outlet
card
capacity
data
Reviews
Reviews
Reviews
Reviews
The Videos comes out very well
I recorded videos without any problems
The videos are of good quality
This Phone has the worst battery ever.
The battery is a deal breaker.
Short battery life!
The cable gets tangled quickly
Firm cable, but flimsy head
Long cable length is a bonus
High capacity memory card
Sufficient capacity to store work files
The capacity is low.

Fig. 3. Partial representation of identified taxonomy

4.3 Results

Evaluation for this paper is complicated by the lack of similar works to compare to. The aim of the paper is not to improve on Aspect Identification systems despite differences with the technique used here with that of other unsupervised aspect extractors. The paper is meant to change review annotation and presentation of Sentiment Analysis.

4.4 Feature Identification

The following formulae, from [8] will be employed for evaluation.

$$Precision = \frac{NumberofCorrect}{NumberofExtracted} \tag{3}$$

$$Recall = \frac{NumberofCorrect}{NumberofTrue} \tag{4}$$

$$F\text{-}measure = \frac{2 * recall * precision}{recall + precision} \tag{5}$$

Evaluating the features identified for the MobilePhone master domain and 5 aspect classes (Table 1):

Table 1. Results of current system aspect identification

Review domain	Precision	Recall	F-measure
Mobilephone	0.64	0.62	0.63
Camera	0.8	0.77	0.785
Charger	0.6	0.552	0.575
Memory	0.65	0.421	0.511
Battery	0.55	0.467	0.505
Box	0.35	0.34	0.345

To compare the system, another evaluation using semEval 14 dataset of laptops (E-Commerce domain) is presented. The performance is compared against Garcia et al.'s paper [6] which uses double propagation with generic opinion words (good and bad) as seeds to generate aspects, and Jose's 2015 submission [13] which uses a Machine Learning approach for the same dataset. The aspect set was limited to top 300 terms (Table 2).

Table 2. Comparison of aspect identification

System	Precision	Recall	F-measure
Baseline	0.443	0.298	0.356
Garcia	0.279	0.444	0.343
Sentiue	0.577	0.441	0.5
Current (top 50)	0.640	0.360	0.461
Current (top 100)	0.600	0.320	0.417
Current	0.323	0.434	0.361

4.5 Taxonomy Evaluation

Tartir et al. [14] have presented some useful metrics for ontology evaluation that could apply to taxonomies as well. The taxonomy is evaluated for class richness and inheritance richness.

Class Richness: Class richness relates to how instances are distributed across classes. It is calculated by dividing the total number of classes with instances by the total number of instances identified.

Inheritance Richness: Average relevant instances per class with instances is measured using the Inheritance Richness formula. It is essentially the average number of classes per sub-tree (Table 3).

Table 3. Evaluation of taxonomy

Domain	Class richness	Inheritance richness
Mobilephone	0.062	6.5

4.6 Annotation Speed-up

To evaluate the decreased classification time, 4 Computer Science Graduates were given 350 sentences each, from the previous laptop domain dataset, to classify without the taxonomy and with the taxonomy. Annotators A and B classified Set1 and Set3 without the taxonomy, and Set2 and Set4 with the taxonomy. Annotators C and D did the opposite. The Sets were given in order to reduce experience bias. "Average" represents the average time without Taxonomy, and "Average(T)" represents average with taxonomy (Table 4).

Table 4. Evaluation of speed-up

Annotator	Set1	Set2	Set3	Set4
A	1	0.87	1.11	0.93
B	1.21	0.90	1.17	0.94
C	0.96	0.95	1.01	1.12
D	0.91	0.99	0.90	1.12
Average	1.11	0.97	1.09	1.12
Average(T)	0.94	0.89	0.96	0.91

Annotator A's classification time of set A (2.52 h excluding breaks) is used as baseline, and the rest are compared to it. The use of taxonomy gives a speed-up of close to 15%. When all the reviews under an aspect are given to an annotator to classify by sentiment polarity, the performance of the annotator improves, partially because they tune themselves to the aspect's review style. A better User Interface is expected to improve annotation speed, over the existing Database querying & spreadsheet based system.

5 Future Work

Further work could be done to upgrade the taxonomy using features of ontology systems, which require relationships to be defined between the hypernym and hyponym. The presentation of sentiment analysis after the sentiment scores are determined will be enhanced if the relationships are found.

Taxonomy creation performance is not a metric that this paper has been judged by. Incorporating strategies to improve presentation and querying can be done to improve the current system. Additionally, the unsupervised aspect identification method could vary. One with higher accuracy could improve the overall system.

6 Conclusion

The paper presents a method to assist annotators in the task of sentiment mining which is growing in importance and value across numerous domains ranging from E-Commerce, Services (Hotels & Restaurants) and Brand Image perception. When an automated system can assist with some of the work, the annotator's job is reduced. This system works because aspect term identification for a particular sentence is binary - either the aspect the sentence is identified under is relevant and maintained, or irrelevant and discarded. This semi-automated annotation task between acquiring the reviews and training can be termed hybrid/semi-supervised because it uses both machine automation and human guidance.

References

1. Balaji, V., Bhatia, M.B., Kumar, R., Neelam, L.K., Panja, S., Prabhakar, T.V., Samaddar, R., Soogareddy, B., Sylvester, A.G., Yadav, V.: Agrotags - a tagging scheme for agricultural digital objects. In: Sánchez-Alonso, S., Athanasiadis, I.N. (eds.) Metadata and Semantic Research, pp. 36–45. Springer, Heidelberg (2010)
2. De Knijff, J., Frasincar, F., Hogenboom, F.: Domain taxonomy learning from text: the subsumption method versus hierarchical clustering. Data Knowl. Eng. **83**, 54–69 (2013)
3. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 231–240. ACM (2008)
4. Ehrig, M., Maedche, A.: Ontology-focused crawling of web documents. In: Proceedings of the 2003 ACM Symposium on Applied Computing, pp. 1174–1178. ACM (2003)
5. Ely, J.W., Osheroff, J.A., Gorman, P.N., Ebell, M.H., Chambliss, M.L., Pifer, E.A., Stavri, P.Z.: A taxonomy of generic clinical questions: classification study. BMJ **321**(7258), 429–432 (2000)
6. Garcıa-Pablos, A., Cuadros, M., Gaines, S., Rigau, G.: V3: unsupervised generation of domain aspect terms for aspect based sentiment analysis. In: SemEval 2014, p. 833 (2014)
7. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
8. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 1st edn. Prentice Hall PTR, Upper Saddle River (2000)

9. Liu, B.: Sentiment analysis and opinion mining. Synth. Lect. Hum. Lang. Technol. **5**(1), 1–167 (2012)
10. Mukherjee, S., Bhattacharyya, P.: Feature specific sentiment analysis for product reviews. In: International Conference on Intelligent Text Processing and Computational Linguistics, pp. 475–487. Springer (2012)
11. Popescu, A.M., Etzioni, O.: Extracting product features and opinions from reviews. In: Kao, A., Poteet, S.R. (eds.) Natural Language Processing and Text Mining, pp. 9–28. Springer, London (2007)
12. Rosenthal, S., Farra, N., Nakov, P.: Semeval-2017 task 4: sentiment analysis in Twitter. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 502–518 (2017)
13. Saias, J.: Sentiue: target and aspect based sentiment analysis in SemEval-2015 task 12. Association for Computational Linguistics (2015)
14. Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P., Aleman-Meza, B.: OntoQA: metric-based ontology quality analysis (2005)
15. Yu, J., Zha, Z.-J., Wang, M., Chua, T.-S.: Aspect ranking: identifying important product aspects from online consumer reviews. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1496–1505. Association for Computational Linguistics (2011)
16. Yu, J., Zha, Z.J., Wang, M., Wang, K., Chua, T.S.: Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 140–150. Association for Computational Linguistics (2011)
17. Zimbra, D., Ghiassi, M., Lee, S.: Brand-related Twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks. In: 2016 49th Hawaii International Conference on System Sciences (HICSS), pp. 1930–1938. IEEE (2016)

History Management for Network Information of IoT Devices

Daeil Jang, Taeeun Kim, and Hwankuk Kim(✉)

Security Technology R&D2 Team, Korea Internet Security Agency, South Korea,
9 Jinheung-gil, Naju, Jeollanam-do, Republic of Korea
{dale,tekim31,rinyfeel}@kisa.or.kr

Abstract. In an Internet of Things (IoT) environment, forensics is commonly used to perform accident analysis through network communication data and the existing memory and logs in a device. Network traffic and memory are volatile data, however, and IoT device logs pose difficulties in information retrieval as opposed to a PC environment due to device and environmental constraints. To do this, we will discuss history management of network information to analyze an accident. History management can be performed on 13 items including IP, firmware version, port number, protocol, service version, and vulnerability information associated with it, and selection of the time and object of infringement can be done by using the Euclidean distance for changeable data.

Keywords: IoT · History management · Network forensics

1 Introduction

Analysis of time-series data is crucial for accident analysis given the high difficulty in determining the point of accident. To do this, analysis is done on forensic artifacts such as volatile data, file systems, logs, and others. In the case of forensics for IoT devices, however, acquisition and analysis of information is difficult due to environmental constraints. For example, memory forensics requires extraction of the memory from the device or physical connection to another device to receive the data; and the size of the memory itself is not large enough. And since the size of storage used by IoT devices is insufficient to accumulate logs, it is almost impossible to acquire data such as logs since the time of the intrusion gets farther away.

This paper discusses management of network information history within an IoT environment at the time of such infringement. Many studies have analyzed and utilized network traffic information, but this one focuses on real-time analysis and detection. If the attack succeeds by bypassing it, however, it must rely on the actions and logs generated by the system. As mentioned above, the information within an IoT environment is not of high quality and quantity, and poses many problems in performing analysis. This study intends to show that the system can be used for infringement by tracking changes in network information in case of infringement by periodically grasping the situation of the network and converting it into time-series data.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 138–145, 2018.
https://doi.org/10.1007/978-3-319-76451-1_14

Section 2 reviews previous research on network information collection and time-series data analysis. Section 3 selects a history management object from network information, and uses the possibility of change and Euclidean distance to examine graph progress following the changes in network information. In Section 4, this information is used to find how to use the system in case of infringement. Finally, Section 5 discusses this study's conclusion and future work.

2 Related Works

Forensics for IoT devices can be divided into network forensics, which includes analysis of traffic and operational information that can be collected at the network level, and device forensics, which separates the embedded log and memory in a device. This study will examine network forensics that can be applied in an IoT environment.

Network forensics monitors and analyzes network traffic for information gathering and legal evidence or intrusion detection purposes. This form of forensics is commonly used for two purposes, generally for handling volatile information related to network traffic. The first is monitoring abnormal traffic on the network and identifying attacks and intrusions. Network-based evidence is the only evidence available for forensic analysis because an attacker can erase all log files on a compromised system. The second can include analysis of network traffic captured as statutory evidence, reanalysis of the transferred files, search for keywords, and parsing of human communications.

2.1 Network Information Collection Tool

A variety of tools are available for gathering information from the internet connected devices, including those for statically collecting network traffic like wireshark [1] or for dynamically collecting information such as Nmap [2], Shodan [3], and Censys [4]. Static collection is a method for extracting the desired information by collecting and processing traffic from network switches and routers. Dynamic collection extracts the desired information based on response values received after sending scan traffic to a specific object or protocol header.

2.2 Analyzing Time-Series Data

The time-series analysis of network information requires finding and searching for patterns in time-series data. A general pattern definition for this is the pattern represented by a time-series graph following the characteristics of its shape [6–8]. Or this study proposes a method to generate segments by slicing the time-series data at a certain time interval and apply the nominal representation to each segment [9], or to represent the pattern as a probability model [10–12].

3 Target Selection for History Management of Network Information

This section examines the data that can be collected from the network information collection system and select the history management object for the forensics. As mentioned above, this is limited to the information that can be collected through network scan, except for the network traffic data that can be collected through the wire shark and other methods.

3.1 Collecting Network Information

A variety of information can be collected through network scanning including IP of the basic target, the port to be scanned, the protocol used by the port, the application, and the version information. The list of collectable information is as follows

- IP, Port, Protocol, Application Name, Application Version, OS Name, OS Version, Firmware Version, Product Name, Product Version, Protocol Header, Protocol Content.

This study uses a ZMap-based customized tool to extract follows information. ZMap is a fastest Internet scanner in the world. This tool consists of Zmap and ZGrab. The ZMap is scanning the network using Syn scan or ICMP scan, and The ZGrab features information on 16 protocols and converted into JSON format including data as follows

- IP, Port, Protocol, Application Name, Application Version, OS Name, OS Version, Firmware Version, Product Name, Product Version, Protocol Header.

Here, the remaining information is extracted using a rule-based fingerprint using the TCP/IP header information for identifying the OS, and a firmware identification technique for each IoT device for firmware identification.

- OS Name, OS Version, Firmware Version, Product Name, Product Version.

3.2 Vulnerability Matching Information

The collected information is used as information for measuring a device's vulnerability. In this case, based on the CVE (Common Vulnerabilities and Exposure) that is publicly known cybersecurity vulnerabilities, the possibility of vulnerability is judged based on the application and version information matching with CVE, OS and version information, firmware and version information. If the CVE vulnerability list includes the version of the application corresponding to the scanned result, the CVE-ID is managed by matching the information. To do this, this study utilizes the CVE crawler, formal vulnerability information analysis and parsing technology, and keyword matching technology.

3.3 Structure of History Management Set for Target Information

When the collected information is matched with that of vulnerability, a set for history management is formed. The basic structure is as follows:

- IP - Date - Product Name (Version) - Firmware Version - OS
  └ Port1 - Protocol - Application - CVE-IDs
  └ Port2 - Protocol - Application - CVE-IDs
  └ Port3…

It first records the date of scanning one IP and identifies the product name and version, firmware version, and OS of the identified device. In the case of a port, a plurality of ports can be opened in one device, information about protocols and applications for each port is collected, and CVE information to be matched is described. The generated history management set manages data collected based on IP-specific dates.

These managed data have various meanings as time-series data. First, the status of connected devices is ascertained. In general, Shodan and Censys, as well as security controls, show only the present situation and the flow of time is not expressed. This can be used to identify devices that are potentially infectious at the time of the infestation. Second, network changes of devices that are assigned IP can be observed. The IoT device uses the same device of various objects due to environmental characteristics except the home IoT. At this time, changes in the same network of the potentially infected device family can be an important artifact for determining the point of infringement in the analysis.

3.4 Analysis of Time-Series Data Using Euclidean Distance

To analyze time-series data, this study generally defines a pattern for time–series information and creates a learning model to generate a learning model for each pattern. The similarity between the generated learning model and the new query sequence is then measured to determine which pattern the new query sequence has similar characteristics to. In the case of the IoT environment, however, little change in the collectible network information means this method is less effective than system resource consumption. For example, a change in the number of open ports per IP, product-firmware version information, access page, and others might have changes in network information, but changes in information are infrequent. This study analyzed the time-series data by using open ports, information on product firmware version, and the Euclidean distance between the access page analysis results. The Euclidean distance between two points in 3-D is calculated as follows.

$$d(p,q) = \frac{1}{1 - \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2 + (z_p - z_q)^2}} \tag{1}$$

p is the current data, q is the previous data, x is the port number, y is the firmware version, and z is the access page analysis result. If this is applied to the IoT environment, the Euclidean distance value 1 between the current data and the immediately preceding data is shown to appear in most sections. In this case, the Euclidean distance is changed to a value other than 1 when the information is changed due to firmware update, new service creation, or infringement. Also, when accessing from a new IP, a value other than 1 is displayed.

Figure 1 shows the Euclidean distance when the information on the index page is changed in a single device and recovered. In this graph, two non-1 values means information changed at that time. Then, second non-1 value is still being maintained. For example, an administrator update a firmware and downgrade or this device was hacked and recover it. Figure 2 shows the Euclidean distance when network information is changed one time in a single device. It means network information is not changed when the information was changed for whatever reason such as firmware update or intrusion. Figure 3 shows the change of network information by using the standard deviation of the Euclidean distance for each device when the infringement accident is spread by the vulnerability in the same device group. Finally, Fig. 4 shows the standard deviation of the Euclidean distance of the devices in the device group when a new device is added to the same group.

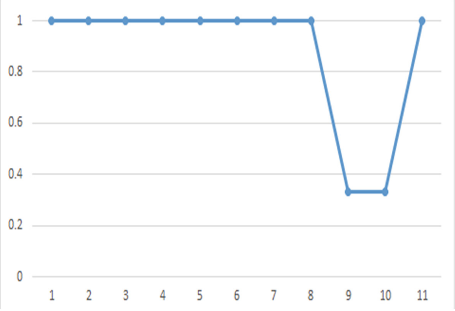


Fig. 1. Recovery after changing device information

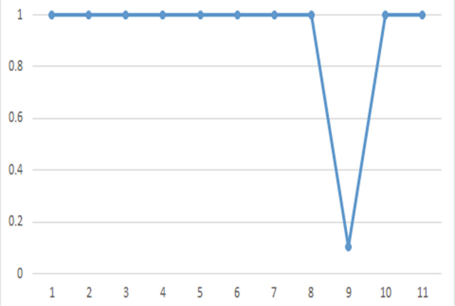


Fig. 2. Continuing after changing device information

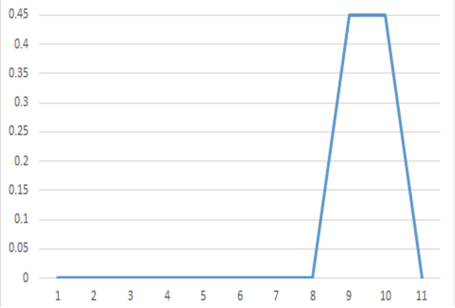


Fig. 3. Propagation spread in same device group

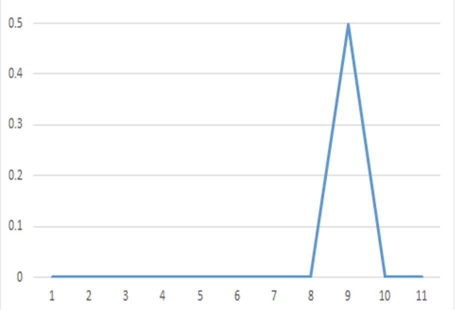


Fig. 4. Add device to device group

4 Scenario Based Utilization Plan

Section 3 discusses the network information of devices that can be collected in the IoT environment, the classification of information that can be changed, and the analysis of time-series data using Euclidean distance of information. Section 4 presents a potential infiltration in the IOT environment and studies how the history management of network information is helpful to accident analysis in case of infringement.

4.1 Utilization of History Management in Infringement by Unauthorized Device in Wireless Network

IoT devices are mostly physically separated and communicate via wireless networks like ZigBee, Z-Wave, Wi-Fi, Bluetooth and others. These wireless protocols have transmission distances ranging from tens to hundreds of meters. This means that it is possible to externally access devices using wireless protocols, which has been proven through the 2013 case of malicious codes deployed in Russia aimed at unsecured wireless networks (with no security setting) using irons and electric kettle [4, 5]. In particular, detection and management of unauthorized devices are crucial in analysis of infringement accidents caused by unauthorized devices, especially when connected to IoT networks and when spoofing is performed to directly connect to devices and attacks.

In this case, based on a global scan for the management network, management of the information is possible through history management of the information of the devices connected to the network. Figure 5 shows when a new device appears in the device group and information on other devices is changed. The standard deviation change in the second time window indicates the addition of the device, that in the third time window indicates network change due to the intrusion, and that in the fourth time window indicates unauthorized device removal.

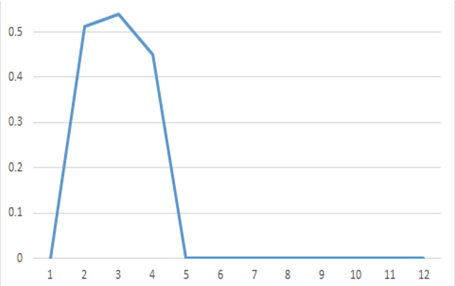


Fig. 5. When unauthorized device is added and intrusion occurs

Based on the corresponding graphs, this study tracks behavioral changes for a window each time based on the change of information in the network in the second, third, and fourth time windows. In this case, by checking the network history of each device, the addition of a new device and change in information of the device group are possible, which can help infer an infringement occurrence time and select an analysis target.

5 Conclusion

In an IoT environment, performing an analysis of infringements requires a lot of time and effort due to limited information. Volatile data such as network traffic requires an especially quick response. So the storage of network information for accident analysis in an IoT environment is critical.

This paper shows that storage of network information in an IoT environment can be used for accident analysis by performing history management on network information in an IoT environment. It can be used to select the point of infringement accident by detecting the change in network information and providing information on the change point through analysis of similarities according to the Euclidean distance at time of collection, with changeable information among network information as the subject. Also, the study confirmed that it can be used to select the analysis target by providing vulnerability information usable in the case of infringement caused by linking network and vulnerability information and the device list having vulnerability at the time of infringement.

Current history management is performed by information collected through dynamic scanning to the device itself, and a method for utilizing history management is suggested. In addition, studying the history management of inter-device communication, which is volatile data, is required as well as needed information on inbound and outbound traffic.

Acknowledgement. This work was supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2016-0-00193, IoT Security Vulnerabilities Search, Sharing and Testing Technology Development).

References

1. Wireshark: Wireshark Foundation. https://www.wireshark.org/
2. Nmap Security Scanner. https://nmap.org/
3. Shodan. https://www.shodan.io/
4. Censys: University of Michigan. https://censys.io/
5. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: fast internet-wide scanning and its security applications. In: Proceedings of the USENIX Security Symposium, August 2013
6. Xianping, G.: Pattern Matching in Financial Time Series Data (1998)
7. Keogh, E.: A fast and robust method for pattern matching in time-series databases. In: Proceedings of the 9th International Conference on Tools with Artificial Intelligence, pp. 578–584 (1997)
8. Keogh, E., Smyth, P.: A probabilistic approach to fast pattern matching in time series databases. In: The Third Conference on Knowledge Discovery in Database and Data Mining, pp. 24–30 (1997)
9. Wang, W., Yang, J., Yu, P.S.: Mining patterns in long sequential data with noise. ACM SIGKDD Explor. **2**, 28–33 (2001)
10. Ge, X., Smyth, P.: Deformable Markov model templates for time-series pattern matching. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, vol. 20, no. 23, pp. 81–90 (2000)
11. Malegaonkar, A., Ariyaeeinia, A., Sivakumaran, P., Fortuna, J.: Unsupervised speaker change detection using probabilistic pattern matching. IEEE Sig. Process. Lett. **13**(8), 509–512 (2006)
12. Khan, B.H.: A Framework for Web-Based Learning. Educational Technology Publications, Englewood Cliffs (2000)

Privacy and Cloud Computing

A Noise Generation Scheme Based on Huffman Coding for Preserving Privacy

Iuon-Chang Lin(✉) and Li-Cheng Yang

Department of Management Information Systems, National Chung Hsing University,
Taichung, Taiwan
iclin@nchu.edu.tw

Abstract. The cloud computing technique rises in these years. Due to cloud computing techniques have some features including low cost, robustness, flexibility and ubiquitous nature. The data in organization will increase immediately. A large number of data can be used on many applications of data analysis involves business, medical and government. But it has some privacy issues, if dealer wants to understand their customer behavior for requirement of marketing, they may publish data into data analysis company, third-party, to analysis. To preserve privacy in database, this paper proposes an efficient noise generation scheme which is based on Huffman coding algorithm. The features of Huffman coding algorithm are a character with lower occurrence frequency has longer code and vice versa. It is suitable to be applied on protecting privacy on database, that tuple with lower occurrence frequency has more noise. The paper presents a noise matrix, a set of noise, which is based on this concept. Although this scheme may lead to data distortion by replace original value, but does not affect to data analysis. In the section of experiments, we consider running time of noise generation with integer number and real number. Overall, this paper shares different concept to perturb original value and propose an efficient data perturbation scheme.

Keywords: Huffman coding · Noise matrix · Numerical database
Privacy preserving

1 Introduction

The cloud computing technique rises in these years. Due to cloud computing techniques have some features including low cost, robustness, flexibility and ubiquitous nature. The data in organization will increase immediately. The term of big data emerged recently that is a collection of dataset which has some features of velocity, volume and variety. Big data can be used on many applications of data analysis involves business, medical and government when the organization generates massive data rapidly. For an instance, when dealer wants to understand their customer behavior for requirement of marketing, they may publish data into data analysis company, third-party, to analysis. However, customer privacy may have leakage crisis, when all transaction data involving personal be published to the third-party, which cannot be trusted. Hence, protecting personal privacy becomes very important. All of privacy disclosure in this research can be simply

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 149–160, 2018.
https://doi.org/10.1007/978-3-319-76451-1_15

classified to two categories. The identity disclosure that adversary is able to match a tuple in a database to an individual. The prediction disclosure that adversary is able to predict the confidential value of an individual. This paper focus on that combination of two or more insensitive attributes can be identified to an individual by an adversary.

In response to a sudden surge in data, the data analysis company may use distributed database system to build their storage environment. Some solutions of dealing big data based on distributed database systems have been proposed. To enhance speed of processing, object-based storage is a choice. Mesnier et al. mentioned an object is a logical collection which is variable size and can be used to store any type of data, such as files, database records, medical images, or multimedia [14]. In practice aspect, there is a merchant produces object storage device which has many features and benefits including high performance, increase storage utilization, keep current investment, and support server/desktop virtualization [21]. In 1998, NoSQL (Not only Structured Query Language) concept has been proposed and redefined in 2009. The main advantages of NoSQL are following: (1) reading and writing data quickly; (2) supporting mass storage; (3) easy to expand; (4) low cost [8]. The data model of NoSQL can be classified to four categories, key-value, column-oriented (column families), document store, and graph databases.

Due to NoSQL database offers high performance and high availability, Tudorica et al. compared performance on several NoSQL databases [16]. More and more companies whose need to deal big data may select NoSQL database as their data storage. But NoSQL database has lack of encryption support. There are some security features can be discussed including authentication, authorization, auditing, client communication, injection and Denial of Service [15]. However, this paper only focuses on preserving privacy on numerical database. We proposed an efficient noise generation approach which is based on noise matrix that can enhance speed of adding noise to original data. Our approach will perturb data to achieve privacy protection but will not reduce data analysis correctness when data perturbation.

The rest of the paper is organized as follows. We review relational literature in Sect. 2. We organize several different solutions for privacy protection and illustrate respectively. And then we illustrate noise generation approach based on Huffman coding in Sect. 3. We use noise generation approach to build a noise matrix, set of noise. After generating noise, we illustrate noise injection procedure. Also, we make the experiments to analysis the running time for proposed approach in Sect. 4. Finally, we summarize the paper in Sect. 5.

2 Related Work

In this section, we review some literature about privacy issues on database. Roughly, the simple idea of privacy preserving is to encrypt entire data before publishing to the service provider. The other idea is adding noise to perturb original value. And then database answers an incorrect result instead of original value to user. We organize several different solutions as the following.

Agrawal and Srikant [2] proposed a concept of privacy-preserving data mining technique is using data distortion. The sensitive data will be perturbed through a random function, and the function returns a distortion value $x + r$ instead original value x where r is a random value. Each original value x has a corresponding noise value r. They classified training data by decision tree which has a growth phase and a prune phase. All of data will be partitioned to same class in growth phase, and will be perturbed, noise injection, by generalized in prune phase. The objective of prune phase is adding noise into original data when the need for data analysis. So they should determine a split point and partition data. And then also need reconstructing original data by random value, data recovery mechanisms. Liu et al. [13] also proposed a solution which based on decision tree. Their proposed decision tree algorithm can be used to classify both the original and the perturbed data.

Agrawal and Aggarwal [1] proposed an Expectation Maximization (EM) Algorithm which is reconstructing distributions at aggregation level. They also proposed a privacy metrics for quantification and measurement, which is based on concept of mutual information between original distribution and estimated distribution. They concern perturbing data and reconstructing distribution, and proposed a robust estimate of original distribution. The proposed algorithm generates the noise value distribution as same as original. Each noise value can be mapped to original value. Reconstructing distribution algorithm with estimated distribution is approaching original destitution, and minimizes information loss.

Dwork et al. [4] proposed a cryptographic protocol for generating noise through Gaussian, and be used for preserving privacy on statistical databases. The proposed ODO protocol (Our Data, Ourselves Protocol) provides an efficient distributed interactive solution via generating noise to prevent malicious participants. The participant queries the database via a privacy mechanism to make sure participants are not faulty. The main solution considers two types of generating noise, Gaussian and scaled symmetric exponential, to generates appropriately distributed random noise. Li et al. [12] proposed Multilevel Trust approach by Gaussian for PPDM (Privacy Preserving Data Mining). The assumption is data owner trusts the data miners at different trust levels and generates differently perturbed copies of the same data for data miner at different trust levels. The data miner has higher trust levels may access perturbed copies at lower than its levels.

Guan et al. [6] proposed an IBE (identity-based encryption) protocol for privacy protection which is a kind of public key cryptosystems, and is based on bilinear maps on elliptic curves. This scheme adopts user's identity information as the public key, and the private key is generated by private key generators (PKGs). The IBE scheme is composed by four algorithms are following: (1) Select a security parameter, and get system parameters and the master key; (2) ID as a public key, and generate private key through PKGs; (3) Encrypt plaintext and get ciphertext; (4) Decrypt ciphertext and get plaintext.

Guo et al. [7] proposed solution based on classical Elgamal homomorphic encryption scheme. Homomorphic encryption allows the particular type of computation to encrypt into the ciphertext, and obtains the encrypted result which is ciphertext of encryption processing on plaintext. According to proposed solution, data are encrypted in key-value pair before publish to database service provider, and construct index. In order to data maintenance, they also proposed data inserting algorithm and data deleting algorithm. Meanwhile, they proposed a query protocol through Paillier encryption scheme. Finally, they implement the proposed algorithms on Berkley DB, a NoSQL database.

Kellaris and Papadopoulos [11] proposed GS scheme with ϵ-differential privacy which is based on grouping and smoothing technique, and focus on one-time publishing of non-overlapping counts. Their scheme focuses on scenario which has privacy issue with aggregation data. Such as social networks may sell user “check-in” summaries to advertising companies, or hospital may provide patient’s prescription aggregates to research unit for some research purposes. The proposed scheme groups the aggregation and smooths them via centroid of group, and minimizes the smoothing perturbation.

Zhang et al. [17–20] proposed a serial noise generation strategy which is based on historical probability. The customer’s privacy information may leak by untrusted service provider record queries from users. They proposed a scenario which customer takes certain actions to protect their privacy without cooperation with server or encryption and decryption on server side. The difference to the mainly solution of noise injection of random noise is that they record historical probability of all users queries. The random noise approach would need a large number of different noises, and the same noise may occur higher frequently. In order to prevent observation by malicious service provider, their idea is to balance noise probability, that noise with higher occurrence probabilities are used less. In contrast, the noise with lower occurrence probabilities are used more.

3 Proposed Scheme

Our scheme is adding noise into original data to achieve preserving privacy. For instance, give a table where has a tuple which has an attribute of age 23. We can revise the value to a range value, 20–30, to prevent adversary observe directly the value in database. The limitation of the paper focuses on statistical database and numerical database, and use data perturbation scheme to protect privacy of database. Although data perturbation approach may replace original data from the measured value, noise. But in generally, numerical database can bear some data distortion in minimum information loss. Minimum data distortion does not affect the data analysis or database operation. In this section we illustrate our concept for noise generation scheme and data perturbation procedure.

3.1 Preliminary

There are many data involving sensitive and insensitive placed in entire database. Intuitively, we only need to protect privacy for sensitive data. We can store directly insensitive data to database because it has not involved personal privacy. However, privacy issues in database have many viewpoints can be discussed. This paper investigates privacy issues from attribute aspect. Some attributes are not considered as sensitive by individual, such as age, weight, education, occupation, and so on. But, the combination of one or more insensitive attributes can be identified each individual, a tuple, by matching those insensitive attributes. The set of one or more attributes by matching insensitive attributes is called quasi-identifier. Quasi-identifier is a set of attributes that can be used to link person if the combination is unique. For an example, give a table which has two insensitive attributes, age and weight. Assume the set of combination of age and weight can be used to identify each individual. The set of combination is a quasi-identifier.

Briefly, the data perturbation approach will use a new value, usually via a measure, to replace original value. To address the identity disclosure for quasi-identifier, this approach will group data and replace original value by centroid of group, average. Although this approach may damage original value and may leads to the problem of data distortion, but will prevent adversary to identify each various individual. However, this paper concern on the problem when the quasi-identifiers are replaced by a centroid of group, that value by grouped may has prediction disclosure issue since grouped value be homogeneous. In other words, this paper considers the disclosure risk of homogeneous sensitive value. We give an example to describe the scope of this research. For simplicity, there is a relation which has a numeric quasi-identifier with two attributes, weight and age, and nine tuples be converted respectively to data point and be placed on two dimensional coordinate systems. We consider a single sensitive value which shows in the Fig. 1. The red points are homogeneous value by grouping measure, and the black points are the other homogeneous value by grouping measure. For example, imagine an adversary knows the weight and age represented by the red points have largest value for weight. It can predict target individual for the sensitive value of red point when adversary uses regroup method which is the loop of dotted lines shown in Fig. 1. Although the value of red point has been homogeneous, but there is a crisis for prediction disclosure, and may leak privacy by regroup method.

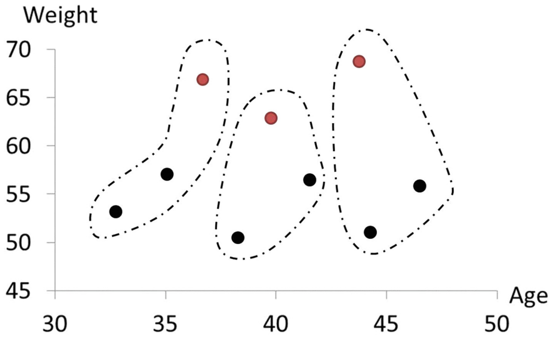


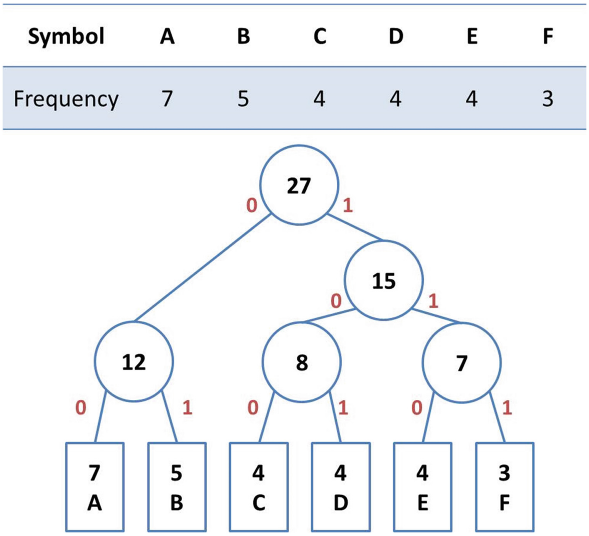
Fig. 1. An illustrative example for prediction disclosure

3.2 Building Noise Matrix

We propose a noise generation scheme that is based on records occurrence probability, and build a set of noise which is called noise matrix in this paper. For a brief description, we consider the quasi-identifier which only has two attributes, and then the inside values of attributes are converted into data point, each data point has one or more homogeneous value, on coordinate systems. In other words, an attribute is indicated to x axis, and another one is indicated to *y* axis. It is indicated to three dimensional coordinate systems when the quasi-identifier has three attributes. We also consider the character of all data point before noise injection. In other words, we should understand how many pattern of data point in the quasi-identifier, for example we can make a pair of age 23 and weight 50 as a unique pattern. Hence, we would classify all patterns, data point, and count the occurrence probability by patterns. The high probability of data point shows that are general in exist, that is mean these data point has more homogeneous value and hard to identify individual. In contrast, the lower probability of data point which will easy be observed by adversary and leak personal privacy.

Our scheme relies on Huffman coding algorithm which proposed by Huffman in 1952 [9]. Huffman coding algorithm is popular on data compression technique [5, 10]. The Huffman coding algorithm has two phases, build Huffman coding tree, Huffman code generation. We can clearly understand number of pattern and the lower probability of data point, easy leakage privacy, by building Huffman coding tree, which can classify to different pattern and count probability of data point. The feature of Huffman coding algorithm is a character has a corresponding code. The character with higher probability has short code. In contrast, the character with lower probability has longer code. According to the feature of Huffman coding algorithm, our noise generation scheme is based on this result of algorithm, and adding more noise into which pattern with lower occurrence frequently to prevent prediction disclosure problem. In contrast, the pattern with higher occurrence frequently may have lower data distortion because there is pattern has more homogeneous value may not leak privacy easily.

We give a simple example to illustrate our idea. All of records are converted to the data point and be placed on coordinate system, which is a quasi-identifier that can be viewed a pattern, we assume each pattern has one or more homogeneous value, such as symbol A, B, and C in the Fig. 2. The Huffman coding tree is built through the occurrence frequency of each symbol. The Fig. 2 defines each symbol has a corresponding frequency respectively, and shows occurrence frequency above the symbol, and the number of internal node of hollow circle is the sum of occurrence frequency from two children. After building Huffman coding tree, then set the weight individually on each branch, that weight on left child is 0 and vice versa the weight on right child is 1. Each symbol has a corresponding Huffman code which we organize into a table shows on Table 1. Our scheme perturbs data through the result of Huffman code, can be viewed noise. Finally, we inject the result of Huffman coding, regarded as noise, into original value directly to perturb data to prevent the problem of prediction disclosure.



Symbol	A	B	C	D	E	F
Frequency	7	5	4	4	4	3

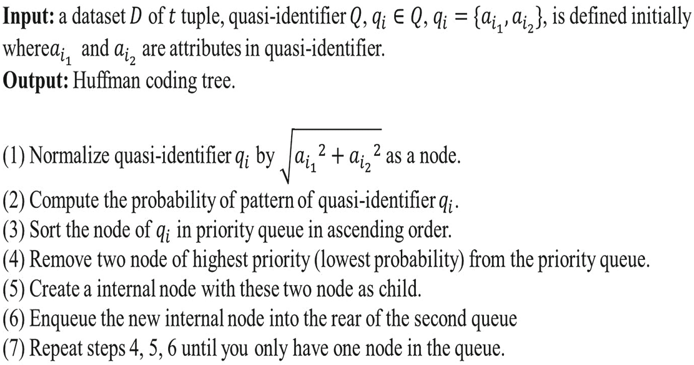
Fig. 2. Huffman coding tree and symbol probability

Table 1. Huffman code

Symbol	A	B	C	D	E	F
Frequency	7	5	4	4	4	3
Code	**00**	**01**	**100**	**101**	**110**	**111**

Some notation is defined to facilitate the discussion. Give a dataset D, and tuple t_i corresponding quasi-identifier $q_i = a_{i_1}, a_{i_2}, q_i \in Q$, where a_{i_1} and a_{i_2} are attributes in quasi-identifier. In order to reduce dimensions in coordinate system with quasi-identifier, this should be normalized before injecting noise to original data. Quasi-identifier is adjusted by Euclidean distances calculation between two values of attribute, $q_i = \sqrt{a_{i_1}^2 + a_{i_2}^2}$. For a brief description, we only concern a quasi-identifier with two attributes. We input a dataset and a quasi-identifier which is defined initially according to our designed algorithm, and output a Huffman coding tree.

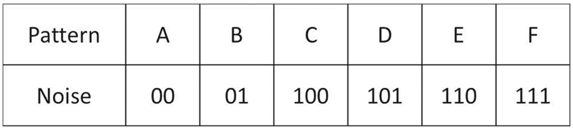
The example mentioned above can be implemented to an algorithm and describe detail for step by step in the Fig. 3 showing. The Huffman code is showed on Table 1 can be converted to a noise matrix which is shown on Fig. 4. Each symbol, the quasi-identifier be viewed a set of various pattern, has a corresponding unique noise value can be injected directly into original value.



Input: a dataset D of t tuple, quasi-identifier Q, $q_i \in Q$, $q_i = \{a_{i_1}, a_{i_2}\}$, is defined initially wherea_{i_1} and a_{i_2} are attributes in quasi-identifier.
Output: Huffman coding tree.

(1) Normalize quasi-identifier q_i by $\sqrt{{a_{i_1}}^2 + {a_{i_2}}^2}$ as a node.
(2) Compute the probability of pattern of quasi-identifier q_i.
(3) Sort the node of q_i in priority queue in ascending order.
(4) Remove two node of highest priority (lowest probability) from the priority queue.
(5) Create a internal node with these two node as child.
(6) Enqueue the new internal node into the rear of the second queue
(7) Repeat steps 4, 5, 6 until you only have one node in the queue.

Fig. 3. Algorithm of Huffman coding tree



Pattern	A	B	C	D	E	F
Noise	00	01	100	101	110	111

Fig. 4. Noise matrix

In order to simplify the process of noise injection, noise generation is based on original data and converted to a noise matrix. The processing of noise injection is directly through original value and noise value corresponding to each other by noise matrix. We use Huffman coding algorithm to generate noise and convert to a matrix to reach the process of efficient data manipulation.

3.3 Data Perturbation Procedure

As the mentioned above, the noise is based on original value to build a Huffman coding tree, then it will convert to a set of noise that we called noise matrix in this paper. Each original value *v* of data point can be corresponded to a noise *r* by noise matrix. This method can be simplified process of original value perturbation, and then the process of noise injection can be more intuitively and easily. After building noise matrix, we will describe the procedure for noise injection. For illustration simplicity, we only consider a quasi-identifier with two attributes. Each data point is mapped to a block in the noise matrix that is visual diagram shown in Fig. 5. The corresponding noise with original value can be injected directly. The noise value has a string of binary, 0 and 1, according to noise matrix that be shown in Fig. 4. We put the string of binary, which is in noise value, into a queue, and add sequentially to original by the function of least significant bit (LSB). The problem of data distortion may significantly reduce due to using LSB function to perturb original value. The replaced value may close to original value, and keep the degree of privacy.

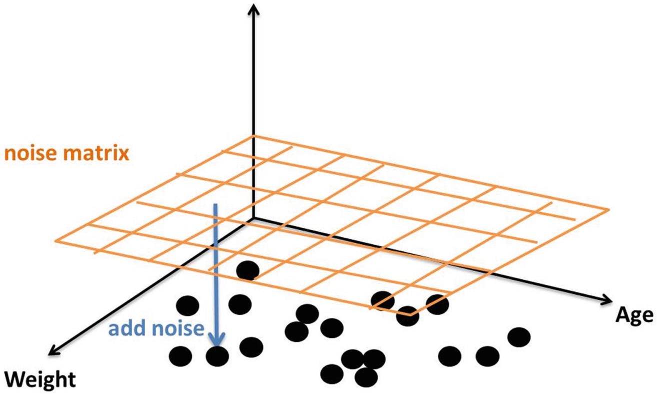


Fig. 5. A diagram for noise injection

We give a simple example to illustrate, assume the original value is $(64)_{10}$, and has a corresponding string of noise value (1010). First, The original of $(64)_{10}$ should be converted to a binary string $(1000000)_2$, and then we put string of binary of noise into a queue $(1010)_{queue}$. We add '1' into string of $(1000000)_2$ and remove a bit from queue. In this case, the length of noise value is 4, so we add 4 into the binary string. The string of binary will be replaced and become to $(1000100)_2$. Finally, we get the result is $(68)_{10}$ by reverse function from binary to decimal.

4 Experiments

The experimental environment is built on a 2.66 GHz processor for Intel Core (TM) 2 Quad CPU and 1.7 GiB of main memory, running the CentOS 6.5 operating system with Linux kernel 2.6. All of algorithms were compiled from the compiler of G++ 4.8.2, and the dataset is provided from UCI Machine Learning Repository [3] which can be used in public. We first describe the background of dataset used in our experiments. The first dataset, 3D Road Network, contains 434,874 instances, with 4 attributes, including OpenStreetMap ID, longitude, latitude, and height in meters. We assume combination of attributes of longitude and latitude is a quasi-identifier where data type of attributes is real number. The second dataset, Census-Income, contains 299,285 instances, with 40 attributes. We assume combination of attributes of age and weeks worked in year is a quasi-identifier where data type of attributes is integer number.

We consider running time of noise generation that calculates per unit time in milliseconds. In order to estimate the precise time, we get the average of 61 times of running time of noise generation. We further discuss the time changes that two datasets manipulate tuples from 500 to 250,000 respectively. Another, we also consider two datasets which has different data type respectively. The first dataset, 3D, which has a numeric quasi-identifier with two attributes of real number. The second dataset, census, which also has a numeric quasi-identifier with two attributes of integer number. In addition, the proposed scheme formalizes quasi-identifier through Euclidean distances calculation, we analysis the result of running time between formalize and non-formalize. The Table 2 shows the result of running time for formalizes quasi-identifier, and the Table 3 shows the result of running time for non-formalizes quasi-identifier.

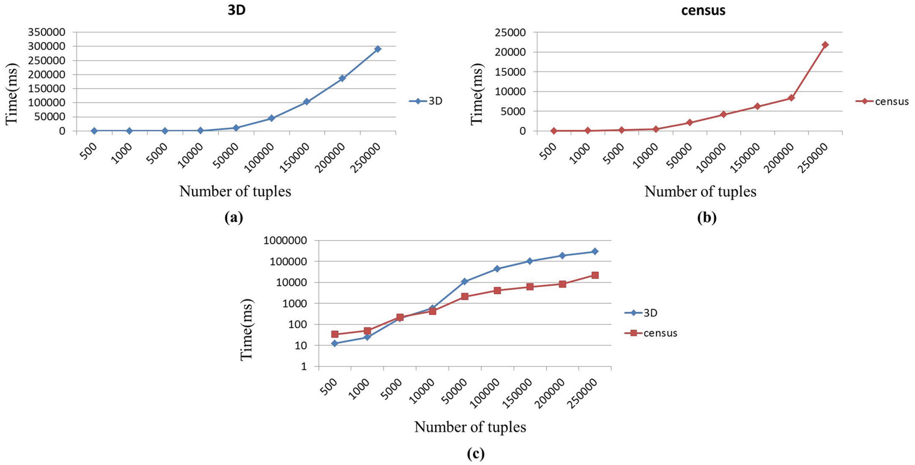
Table 2. The running time of noise generation for formalized quasi-identifier

Tuples	500	1000	5000	10000	50000	100000	150000	200000	250000
3D	12.45902	24.04918	191.2295	584.8197	11070.23	44543.1	103439.7	185823.6	290646.6
Census	33.85246	49.62295	219.3443	427.2459	2097.443	4155.639	6208.279	8319.098	21837.38

Table 3. The running time of noise generation for non-formalized quasi-identifier

Tuples	500	1000	5000	10000	50000	100000	150000	200000	250000
3D	12.09836	23.90164	188.5574	578.5738	10953.57	44367.41	103080.4	185560.3	290231.9
Census	34.93443	47.14754	217.8197	430.3443	2110.836	4177.328	6301.492	8309.574	23385.74

The difference between whether formalized quasi-identifier may leads to a little calculation error. The experimental results show the running time is more close to each other. However, the running time of noise generation for formalized quasi-identifer is higher than non-formalized quasi-identifer. This is because the step of formalized quasi-identifier should more time to manipulate data. Figures 6 and 7 show the running time is incremented with the increasing data, respectively represent formalized and non-formalized. In addition, the beginning of running time of integer is slower than real number, that is shown on Figs. 6(c) and 7(c). Furthermore, since attributes of dataset of census are more than 3D, where the census has 40 attributes but 3D has 4 attributes. Hence, it leads to the beginning of running time of census is slower than 3D, in 500 tuples and 1,000 tuples.

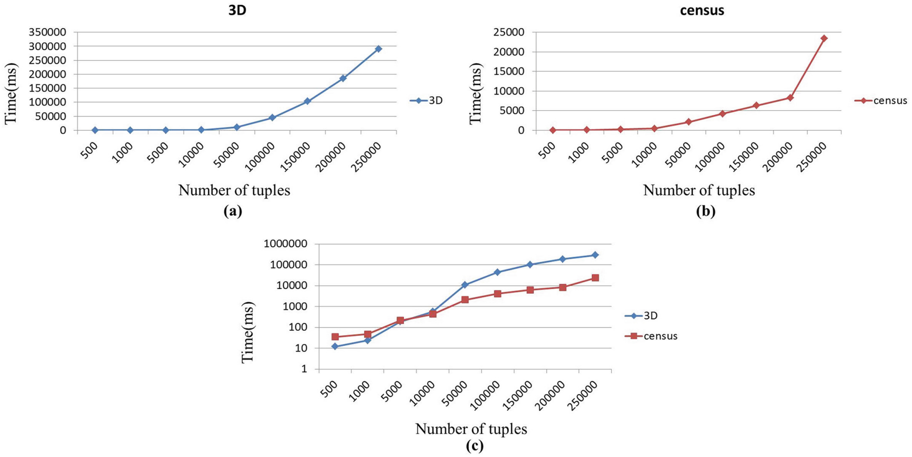


(a)

(b)

(c)

Fig. 6. The trend of running time of noise generation for formalized quasi-identifier



(a)

(b)

(c)

Fig. 7. The trend of running time of noise generation for non-formalized quasi-identifier

5 Conclusions

The use of information and Internet technologies as teaching and learning tools is now rapidly expanding into education. Electronic learning (e-learning) is one of the most popular learning environments in the information age. Indeed, e-learning extends traditional learning paradigms into new dynamic learning models through computer and Web technologies [2].

References

1. Agrawal, D., Aggarwal, C.: On the design and quantification of privacy preserving data mining algorithms. In: ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 247–255 (2001)
2. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: ACM SIGMOD International Conference on Management of Data, pp. 439–450 (2000)
3. Asuncion, A., Newman, D.: UCI Machine Learning Repository. University of California, Irvine (2007). http://archive.ics.uci.edu/ml/
4. Dwork, C., Kenthapadi, K., McSherry, F.: Our data, ourselves: privacy via distributed noise generation. In: Advances in Cryptology - EUROCRYPT 2006, vol. 4004, pp. 486–503 (2006)
5. Gonciari, P.: Variable-length input Huffman coding for system-on-a-chip test. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **22**(6), 783–796 (2003)
6. Guan, S., Zhang, Y., Ji, Y.: Privacy-preserving health data collection for preschool children. Comput. Math. Methods Med. **2013**, 1–5 (2013)
7. Guo, Y., Zhang, L., Lin, F., Li, X.: A solution for privacy-preserving data manipulation and query on NoSQL database. J. Comput. **8**(6), 1427–1432 (2013)
8. Han, J., Haihong, E., Le, G., Du, J.: Survey on NoSQL database. In: International Conference on Pervasive Computing and Applications, pp. 363–366 (2011)

9. Huffman, D.: A method for the construction of minimum redundancy codes. In: The IRE, vol. 27, pp. 1098–1101 (1952)
10. Kavousianos, X.: Optimal selective Huffman coding for test-data compression. IEEE Trans. Comput. **56**(8), 1146–1152 (2007)
11. Kellaris, G., Papadopoulos, S.: Practical differential privacy via grouping and smoothing. In: ACM International Conference on Very Large Data Bases, pp. 301–312 (2013)
12. Li, Y., Chen, M., Li, Q., Zhang, W.: Enabling multilevel trust in privacy preserving data mining. IEEE Trans. Knowl. Data Eng. **24**(9), 1598–1612 (2012)
13. Liu, L., Kantarcioglu, M., Thuraisingham, B.: Privacy preserving decision tree mining from perturbed data. In: International Conference on System Sciences, vol. 5, pp. 1–10 (2009)
14. Mesnier, M., Ganger, G., Riedel, E.: Object-based storage. Commun. Mag. **41**(8), 84–90 (2003)
15. Okman, L., Gal-Oz, N., Gonen, Y., Gudes, E., Abramov, J.: Security issues in NoSQL databases. In: IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 541–547 (2011)
16. Tudorica, B.G., Bucur, C.: A comparison between several NoSQL databases with comments and notes. In: International Conference 10th Edition: Networking in Education and Research, pp. 1–5 (2011)
17. Zhang, G., Liu, X., Yang, Y.: Time-series pattern based effective noise generation for privacy protection on cloud. IEEE Trans. Comput. **PP**(99), 1 (2014)
18. Zhang, G., Yang, Y., Chen, J.: A historical probability based noise generation strategy for privacy protection in cloud computing. J. Comput. Syst. Sci. **78**(5), 1374–1381 (2012)
19. Zhang, G., Yang, Y., Liu, X., Chen, J.: A time-series pattern based noise generation strategy for privacy protection in cloud computing. In: IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 458–465 (2012)
20. Zhang, G., Yang, Y., Yuan, D., Chen, J.: A trust-based noise injection strategy for privacy protection in cloud. Softw. Pract. Exp. **42**(4), 431–445 (2012)
21. EZCloudStor OSD (Object Storage Device): Have Your Own Amazon S3 on Premise. EZ Cloud Tech. http://www.ezcloudtech.com/en/product/ez_osd.php

Privacy-Preserving Outsource Computing for Binary Vector Similarity

Dan Yang[1], Yu-Chi Chen[2](✉), and Shaozhen Ye[1]

[1] College of Mathematics and Computer Science, Fuzhou University,
Fuzhou, People's Republic of China
s1056042@mail.yzu.edu.tw, yeshzh@fzu.edu.cn

[2] Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan
wycchen@saturn.yzu.edu.tw

Abstract. The preservation of privacy has become a widely discussed topic on the Internet. Encryption is an approach to privacy; however, to outsource computing to an cloud service without revealing private information over encrypted data is difficult. Homomorphic encryption can contribute to it but is based on complicated mathematical structures of abstract algebra. We propoase a new scheme for securely computing the similarity between binary vectors through a cloud server. The scheme is constructed from ciphertext policy attribute based encryption and garbled circuits rather than homomorphic encryption. Attribute based encryption provides the access power, which is a necessary primitive in our scheme. Moreover, for computing over encrypted data, we rely on garbled circuits to handle secure outsourcing and to avoid the use of homomorphic encryption.

Keywords: Privacy preservation · Outsourced computing · Data search
Garbled circuit

1 Introduction

Cloud storage is a noteworthy trend in cloud computing and numerous scholars have published research on cloud storage in the last few years. In particular, some IT companies have developed storage services such as Dropbox, iCloud, and SkyDrive. One goal of such storage services is to provide extra space for mobile devices, and another is to maintain synchronization and consistency among multiple devices. Because cloud storage offers great convenience, many users grow accustomed to storing and accessing their data in the cloud.

A drawback of cloud storage is that users must fully trust the cloud server. A challenge of privacy and confidentiality is how to protect sensitive or private data while utilizing the efficient search function of the cloud [1]. Intuitively, secure encryption is a straightforward method for providing data confidentiality; a user must encrypt private data before uploading them to a server. In the future, users may wish to access specific portions of those encrypted data but cannot reveal any private key to the server. Typically, the user must recall all data from the server and then select the required subset.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 161–169, 2018.
https://doi.org/10.1007/978-3-319-76451-1_16

The encrypted data are unreadable random strings, and thus the server cannot directly search for data based on his queries. This solution is trivial and inefficient because the server is merely a storage server and cannot increase efficiency. A user who desires one item must retrieve all items from the server. Keyword-searchable encryption [2, 3] (SE, for short) is introduced to enable efficient searches of encrypted data. In keyword-searchable encryption, a sender uses a receiver's public key to encrypt data and corresponding keywords, and then sends them to the cloud. Subsequently, the receiver can use its private key to generate the trapdoor of search keywords, and the trapdoor output is sent to the cloud. Finally, the cloud can test the keyword ciphertext and trapdoor. If the keyword ciphertext and trapdoor match, the cloud returns the corresponding encrypted data.

Privacy-preserving outsourced computing with binary vector similarity (PPOS) is an application of cloud storage. The system is composed of four entities (user A, user B, the sharing cloud, and the searching cloud) and the scenario is somewhat similar to that of SE. User A extracts the features from the original data and then uploads the encrypted data to the sharing cloud and the encrypted features to the searching cloud. User B generates a search request from the searching features and sends a search token (similar to the trapdoor) to the searching cloud. Subsequently, the searching cloud server returns a result that reflects the similarity between the search features and original features (a.k.a L2-norm). User B applies the result to retrieve the data from the sharing cloud server. Notably, PPOS is slightly different from SE because SE we care the 100% match of keyword, whereas PPOS only care similarity. For security, the sharing cloud server and searching cloud server are collusion-free, and thus some known attacks against SE cannot be used against PPOS.

Zhang et al. proposed a PPOS scheme [4, 5] based on ciphertext-policy attribute based encryption (CP-ABE) [6–8], additive homomorphic encryption (HE) [9], and garbled circuits [10–13]. We briefly summarize the main idea of their scheme as follows. For simplicity, we delineate only the interactions among user A, user B, and the searching cloud server.

- User A generates a public key, a secret HE key, and a garbling key, and then runs key generation for CP-ABE, sets the access structure, and gives user B an access key (notably, user B can perform decryption of CP-ABE for one ciphertext if his access key matches the attributes).
- User A uploads the ciphertexts of $\mathrm{key}_{\mathrm{HE}}$ and the garbling key encrypted by CP-ABE.
- User A generates the XOR(x_A(k), `) garbled circuits for each dimension k of x_A, and the output is the HE ciphertext consisting of 0 s and1 s.
- User B fetches the ciphertext of CP-ABE, decrypts it to obtain the key, and generates the garbled input from the garbling key and his search request x_B.
- Finally, the searching cloud server receives the garbled input uploaded by user B and uses it to obtain HE.E(XOR(x_A(k), x_B(k))) in each dimension. Subsequently, that server computes the summation to obtain HE.E(d(x_A(k), x_B(k))).

Contributions

Our starting point is that we found their improved scheme works with two individual computations (including XOR operations in garbled circuit and summation in HE). The concrete goal is to get rid of the use of HE because garbled circuits can be used to render computation. Hence, we propose a new scheme based on CP-ABE and garbled circuits. The techniques that define our scheme are described as follows:

- User A generates a specific circuit that hardcodes user A's input (feature value) and computes summation of XOR(x_A, .) for each feature dimension; the input of the specific circuit is exactly the search request of user B. This action combines the two individual computations of this scheme into only one.
- User A uploads the garbled circuit of the specific circuit and the CP-ABE ciphertext of the garbling key to the searching cloud server.
- User B can decrypt the CP-ABE ciphertext to obtain the garbling key. User B applies the garbling key to generate the garbled input for his search request, and then sends the garbled input to the searching cloud server.
- Finally, the searching cloud server evaluates the garbled circuit and input and then returns the "plain" evaluation result to user B.

Some minor challenges may require attention but are quite simple to overcome. If we want to preserve the "output privacy" against the searching cloud server, we can trivially set the server to deliver the final result as a ciphertext. In addition, as is known, the entire garbling key may be long, and thus we can deliver a short pseudorandom generator seed instead which is used to generate a long garbling key through a pseudorandom generator. This is a standard technique for achieving compression.

Organization. The remainder of this paper is organized as follows. Section 2 explains some preliminaries including garbled circuits and CP-ABE. Section 3 presents a PPOS framework. In Sect. 4, we introduce the proposed scheme, and in Sect. 5, gives some discussions. Section 6 concludes this paper.

2 Preliminaries

2.1 Garbled Circuits

Garbled circuits were originally designed by Yao [10–13] for two-party computation. A garbled circuit scheme consists of a pair of algorithms (GarbleC and Eval). In terms of high-level functionality, GarbleC is the circuit procedure and Eval is the corresponding result evaluation procedure. Each input and output wire w of the circuit is associated with two labels, lkey_0^w and lkey_1^w, which correspond to the bit-values $\mathrm{b} \in \{0,1\}$. Finally, a garbled circuit $\tilde{\mathrm{C}}$ is applied to blind the evaluation by given the garbling inputs. For security, given garbled circuit and input, it does not reveal anything except for the evaluation result of the output. A garbling scheme usually consists of the two algorithms described as follows.

- $(\tilde{\mathrm{C}}, \{\mathrm{j}, \mathrm{b}, \mathrm{lkey}_\mathrm{b}^{in,j})\} \leftarrow \mathrm{GarbleC}(1^\lambda, \mathrm{C}, \{\mathrm{i}, \mathrm{b}, \mathrm{lkey}_\mathrm{b}^{\mathrm{out},i}\})$: as input, GarbleC takes a security parameter λ, a circuit C, and a set of labels $\mathrm{lkey}_\mathrm{b}^{\mathrm{out},i}$ for all output wires $\mathrm{i} \in \mathrm{out(C)}$ and b ∈ {0,1}. We denote the sets of input and output wires by inp(C) and out(C), respectively. Then it outputs a garbled circuit $\tilde{C}$ and a set of labels $\mathrm{lkey}_b^{in,j}$ for each input wire j ∈ inp(C) and b ∈ {0,1}.

- $(\mathrm{lkey}^{\mathrm{out},1}, \mathrm{lkey}^{\mathrm{out},2}, \ldots) = \mathrm{Eval}(\tilde{C}, (\mathrm{lkey}^{\mathrm{in},1}, \mathrm{lkey}^{\mathrm{in},2}, \ldots))$: Given a garbled circuit $\tilde{C}$ and a sequence of input labels $\mathrm{lkey}^{\mathrm{in},j}$, Eval outputs a sequence of output labels $\mathrm{lkey}^{\mathrm{out},i}$. Intuitively, if the input labels correspond to some input x then the output labels should correspond to y = C(x).

Without loss of generality, we describe the circuit garbling scheme as the following four algorithms.

- **KeyGen(1^λ)**: generates the garbling key
- **GarbleC(key, C(`))**: generates the garbled circuit $\tilde{C}$
- **GarbleInp(key, x)**: generates the garbled input $\tilde{x}$
- **Eval**($\tilde{C}$, $\tilde{x}$): evaluates the result C(x).

2.2 CP-ABE

Ciphertext policy attribute based encryption [6–8] is a universal and safe access control protocol based on ciphertext policy, referred to as **CP-ABE**. In CP-ABE, when data are encrypted by an access control policy, the $\mathrm{key}_{\text{CP-ABE}}$ consists of the Boolean values of the attributes and the Boolean operators (AND/OR). The $\mathrm{key}_{\text{CP-ABE}}$ corresponds to a set of attributes {A}; therefore, only a user that satisfies the attributes of {A} can do decryption successfully.

The following algorithms exist in CP-ABE.

- **Setup**: Given only the implicit security parameter, it outputs the public parameters P and a master key K_M.
- **Encrypt(P, M, AS)**: It takes the public parameters P, a message M, and an access structure AS as inputs. We assume that the ciphertext implicitly contains AS. Subsequently, M is encrypted to the ciphertext CT such that only the user possesses A, the set of attributes that match the access structure AS, can decrypt M.
- **Key Generate(K_M, A)**: It takes the master key K_M and a set of attributes A as input and then generates a private key K_S.
- **Decrypt(P, CT, K_S)**: The public parameters P, the ciphertext CT (which contains an access structure AS), and the private key K_S as input. Because K_S is generated by the set of attributes A, if A matches AS, the algorithm decrypts CT and returns M.

3 System Framework

The system framework is an outsource computing framework of data sharing and searching with privacy-preserving. Figure 1 describes the framework, focusing on two objectives:

1. Sharing data with selectively access delegation
2. Searching data without extra computing overhead

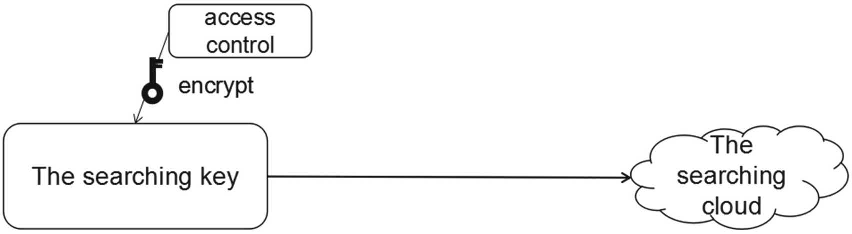


Fig. 1. Key generation

To achieve these objectives, our framework comprises three parts: key generation, data uploading, and data searching.

3.1 Key Generation

The framework enables a user to share his or her data selectively with someone who has access delegation. When user A registers, the system generates his ***searching key*** to be used for data uploading and searching. Subsequently, to prevent others from accessing the searching key, user A can protect the key through access control, as illustrated in Fig. 1. In other words, user A should encrypt the key with access control rules and upload the encrypted key to the cloud so that only the designated user can access the searching key from the cloud.

3.2 Data Uploading

Before being uploaded, user A's data must be preprocessed. The keywords of the data must be defined automatically or by user A. The keywords may be relevant to features

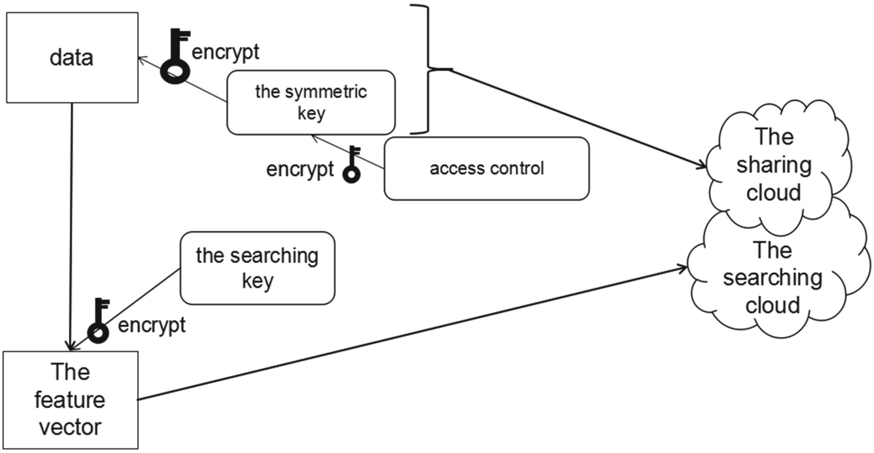


Fig. 2. Data uploading

such as timestamps, document titles, file formats, media content and so on. These keywords are transformed into a binary feature vector.

User A uses symmetric encryption to encrypt the data and uploads the symmetric key (encrypted through access control) as well as the data to the sharing cloud. At the same time, the encrypted binary feature vector is uploaded to the searching cloud, as depicted in Fig. 2.

3.3 Data Searching

When user B plans to search the data of user A, user B must generate the feature vector to describe what he requires. This feature vector is referred to as the searching vector. If the search user B conforms to the rules for access control set by the data owner A,

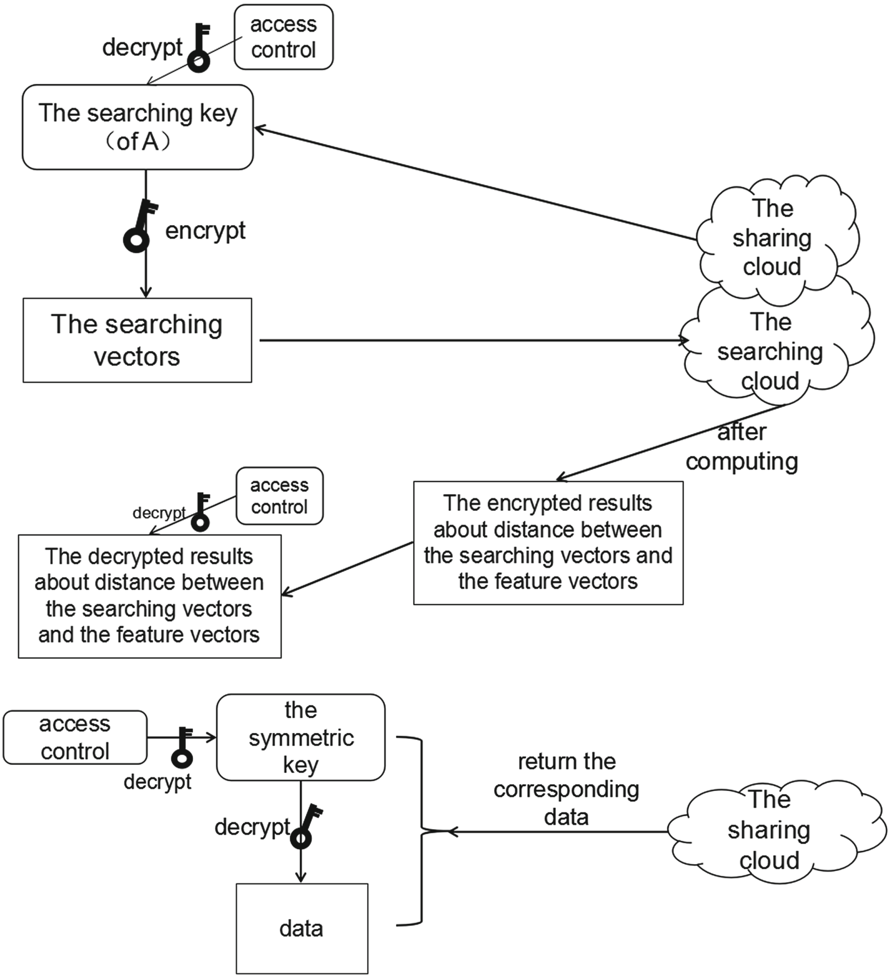


Fig. 3. Data searching

user B is permitted to obtain the searching key of user A from the cloud. Thus, user B can use the searching key to generate an encrypted searching vector.

After that, user B uses the encrypted searching vector to search data. The cloud computes similarity as the distance between the searching vector of user B and the feature vector of all user A's data without any further interactions with user A. The cloud then returns the result to user B. Notably, the result is ciphertext that can be decrypted by user B.

Finally, user B can discover the specified data, referring to the result decrypted through the searching key of A, as illustrated in Fig. 3. As a result, the cloud manages the entire computing process without any plain knowledge of the data or searching content.

4 Proposed Scheme

The essential component of the system is the capacity to search encrypted data. To address this, we propose a new scheme based on garbled circuits that compute the similarity between the searching vector and the feature vector. This section describes how to encrypt and decrypt the vectors and obtain the similarity results.

User A

1. Use KeyGen(1^λ) to obtain the key. When a new user registers, the system must generate that user's searching key.
2. Obtain and upload ABE.E(key). If user A wants to delegate access to another user through access control, he uses CP-ABE to encrypt the key. Subsequently, ABE.E(key) is uploaded to the cloud.
3. Use GarbleC(key, C(`)) to obtain and upload $\tilde{C}$. In these calculations, C(`) is ADD(XOR(inputA, `).

In step 3, user A hardcodes the feature vector as $\mathbf{input_A}$ in a specific circuit for each dimension.

User B

1. Fetch ABE.E(key). If user B plans to get the distance between the vectors, she needs achieve the access right from A. More precisely, user B must fetch ABE.E(key) from the cloud.
2. Decrypt ABE.E(key) to obtain the key. If the attributes of user B match the rule set by user A, she decrypts ABE.E(key) successfully and obtains the key.
3. Use GarbleInp(key, x) to obtain and upload $\tilde{x}$. User B uses the key to translate the searching vector x to $\tilde{x}$ as the input of $\tilde{C}$ and send it to the cloud.

Cloud

1. Use Eval($\tilde{C}$, $\tilde{x}$) to obtain C(x). After receiving $\tilde{x}$, the searching cloud server evaluates $\tilde{x}$ with $\tilde{C}$ to obtain the result of the distance between vectors, and then sends the

distance to user B (for output privacy, the cloud will return the ciphertext version instead) (Fig. 4).

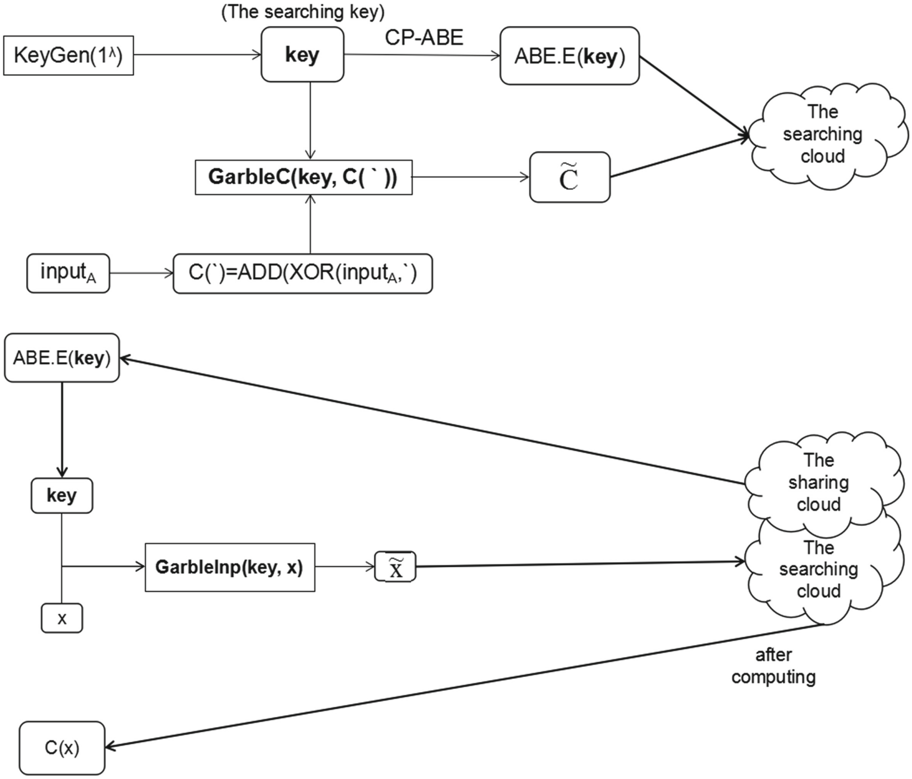


Fig. 4. Sketch of the scheme

5 Discussion

We propose a PPOS scheme where data searching under ciphertext is located in the third-party data center. Generally, implementing encrypted data searching usually relies on HE to compute similarity. However, HE usually incurs excessive computational overhead. Consequently, our scheme is based on only access control and garbled circuits to not only protect the data in the cloud from access by unauthorized users but also enable the data to be searched by authorized users without disclosure.

The scheme is unavoidably affected by the shortage of garbled circuits. Obviously, the garbled circuits cannot be used more than once. Reusable garbled circuits may be required, and this is a possible direction for future research.

6 Conclusion

This paper describes a scheme of PPOS from weak assumptions. The proposed scheme is based on CP-ABE and garbled circuits. In contrast to some previous schemes, our scheme does not use HE, and we apply garbled circuits as primitives to handle secure outsource computation.

Acknowledgements. This work was partially supported by the Innovation Center for Big Data and Digital Convergence, Yuan Ze University, and the Ministry of Science and Technology of Taiwan under grant no. 106-2218-E-155-008-MY3. We also acknowledge Wallace Academic Editing for editing this manuscript.

References

1. Benaloh, J., Chase, M., Horvitz, E., et al.: Patient controlled encryption: ensuring privacy of electronic medical records, pp. 103–114 (2009)
2. Ryu, E.K., Takagi, T.: Efficient conjunctive keyword-searchable encryption. In: International Conference on Advanced Information Networking and Applications Workshops, pp. 409–414. IEEE (2007)
3. Chen, Z., Wu, C., Wang, D., et al.: Conjunctive keywords searchable encryption with efficient pairing, constant ciphertext and short trapdoor. In: Intelligence and Security Informatics, pp. 176–189. Springer, Heidelberg (2012)
4. Zhang, L.: Privacy-preserving computing and applications. Ph.D. thesis, Tsinghua University (2014)
5. Zhang, L., Jung, T., Liu, C., et al.: POP: privacy-preserving outsourced photo sharing and searching for mobile devices. In: IEEE International Conference on Distributed Computing Systems, pp. 308–317. IEEE (2015)
6. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Computer Society (2007)
7. Lai, J., Deng, R.H., Li, Y.: Fully secure cipertext-policy hiding CP-ABE. In: International Conference on Information Security Practice and Experience, pp. 24–39. Springer-Verlag (2011)
8. Kumar, S.N.: Cryptography during data sharing and accessing over cloud. Int. Trans. Electr. Comput. Eng. Syst. **3**(1), 12–18 (2015)
9. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: International Conference on Theory and Application of Cryptographic Techniques, pp. 223–238. Springer-Verlag (1999)
10. Garg, S., Lu, S., Ostrovsky, R., et al.: Garbled RAM from one-way functions. In: STOC (2015)
11. Yao, C.C.: How to generate and exchange secrets. In: Symposium on Foundations of Computer Science, 1986, pp. 162–167. IEEE (1986)
12. Gentry, C., Halevi, S., Lu, S., et al.: Garbled RAM revisited. In: International Workshop/Conference on Theory and Application of Cryptographic Techniques, pp. 405–422. Springer, Heidelberg (2014)
13. Huang, Y., Evans, D., Katz, J., et al.: Faster secure two-party computation using garbled circuits. In: Usenix Conference on Security, p. 35. USENIX Association (2011)

Strategies to Improve Auditing Performance and Soundness for Cloud Computation

Shin-Jia Hwang(✉) and Tsung-Lin Li

Department of Computer Science and Information Engineering,
Tamkang University, Tamsui, New Taipei City 251, Taiwan, R.O.C.
sjhwang@mail.tku.edu.tw, 600410962@s00.tku.edu.tw

Abstract. Since the cloud computation auditing becomes important recently, Wei et al. proposed their cloud computation auditing scheme. However, they assume that the cheating adversary always gives the random response for the auditing challenges. This assumption is impractical. When only a small part of adversary's response is random, the number of challenges is increased dramatically. Then the auditing load becomes so heavy that the auditor cannot give the auditing results in reasonable time. Moreover, the probability of finding out incorrect computed results cannot reach that the users want. To improve the on-line audit performance or probability, the off-line easy-auditor improving strategy, the function-based improving strategy, and mixed strategy are proposed, respectively. Utilizing the off-line computation concept and the cloud computation server help, the online audit performance, and the audit probability will be improved.

Keywords: Merkle hash trees · Cloud auditing · Cloud computing
Cloud storage · Digital signature schemes

1 Introduction

For limited computing power, cloud users need to outsource the computation problem to cloud computing servers to obtain the results. It is convenient for cloud users, but the computation delegation may suffer the wrong result or cheating by cloud computing servers. Cloud users need the verification agencies (VA for short) to audit the results [1–5]. Since the VA should response the auditing results in reasonable time, the audit efficiency of the computation result becomes the next topic to be focused on.

Due to VA's limited computing power, the online audit of the computation results is slow and the probability to find out the wrong results/cheat is not significant enough. So, the performance of online auditing and the probability to successfully identify the wrong results/cheat by online auditing are important issues in computation outsourcing schemes.

For computation outsourcing services, many computation outsourcing schemes [6–13] are proposed to satisfy correctness and soundness requirements. Correctness means that all of the computed results are correct by following the cloud computing service delegation procedure; Soundness means the cheat of cloud computation server or wrong computed results can be find out with a reasonable probability the cloud users

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 170–182, 2018.
https://doi.org/10.1007/978-3-319-76451-1_17

need. Here the probability of finding out the wrong results/cheat is the key issues. Besides, the performance of online auditing should be improved to shorten the response time of online audit [14]. Therefore, We et al. [13] proposed their audit procedure for not only the cloud storage but also the computed results.

To increase the probability finding out the wrong computed results or cheating, the adversary always gives the random answer for the challenging queries in [13]. Thus the auditor needs only 33 challenging data to reduce the probability of finding out the wrong results or cheating. However, in practical situations, the most powerful adversary is the cloud computation servers. The most of computed results might be correct but only small part is wrong or cheating. In this situation, among the challenging results, some responses are correct and some responses are random answer. It is possible that all the responses are correct. To enhance the audit effect, the size of challenging data should be increased dramatically. For the large size of challenged data, the auditor might not online perform the audit task in reasonable time.

To improve the efficiency of online audit and the probability finding out the wrong computed results/cheating, some new audit strategies are proposed. In the following section, our system model, assumptions, and security and performance requirements are described. Section 3 is our proposed data strategies. The last is our conclusion.

2 Preliminaries

2.1 Cloud Computing System Model, Assumptions, and Security Requirements

There are three main entities: Cloud Users (CUs, for short), Cloud Computing Servers (CCSs, for short), and Verification Agencies (VAs, for short). CUs are the data owners and propose the cloud computing service requests to CCSs. CCSs are the cloud computing services provider because of their huge computing power on Internet. CCSs also provide cloud storage services for CUs to store the data and computed results. A VA should be trusted by some CUs and accepts the auditing delegation from CUs to audit the computed results stored in CCS. The computing power of VA may be stronger than CUs.

In the system models, some assumptions are described below. First, before delegating CCS's cloud computing services, CU has uploaded CCS the data for the cloud computing requests. Those uploaded data is also signed by the data owner, CU, using the public key based signature schemes to guard against any modification on those uploaded data. Each one of those uploaded data has its unique indexes. Next, assume that the computing power of the CCS is much stronger than the computing power of CUs or VAs. Moreover, the storage size of CCS is also much larger than the storage size of CUs or VAs. In our model, the CCSs and CUs have their certificated public keys and the corresponding private keys for the underlying signature schemes.

The cloud computing service delegation procedure is roughly described below. First of all, some CU sends CCS the cloud computation request and delegates CCS to perform the delegated computation functions on CU's uploaded data. CU assigns one or more computation functions for the cloud computing requests to CCS. After receiving the cloud computation requests, CCS computes and stored the computed results. CCS also generates and responds CCS's signature on the computed results to CU as evidences.

The auditing procedure is briefly described here. After obtaining the CCS's signature on the cloud computed results, the CU may delegate some VA the verification of the cloud computed results. To delegate the auditing task, CU sends VAs the CU's signature on the uploading data, the CCS's signature on the cloud computed results, the related data and cloud computing functions. VA helps CUs validate the correctness of the cloud computed results. Due to the limited computation power, the VA randomly selects some data indexes for cloud computing service requests and sends those indexes to the CCS. According to the received selected indexes, CCS sends VA the CU's uploading data and the CCS's computed results. VA first validates the integrity and source of the received uploading data using CU's signature. VA validates the integrity and source of the received cloud computed results using CCS's signature. Then VA checks the correctness of cloud computed results using the cloud computing functions and uploading data. If all the cloud computed results of the randomly selected data are correct, then VA reports that the CCS computed results may be correct; otherwise, VA reports that the part of the CCS computed results are incorrect.

The first security requirements of the cloud computing services is the correctness, the second is soundness, and the third is efficiency. The correctness means that the computation results are all correct if following the step of ours strategies. Soundness means that the probability of finding out the wrong computed results or cheat is smaller than the cloud users wish. The efficiency is related to the audit performance. The basic way to audit the computed results is that the auditors perform the delegated functions on the chosen delegated data and check whether or not the computed results by auditors are the same as the results by the cloud computation servers. The auditor must give the on-line audit response in reasonable time. Moreover, any improved on-line auditing method should be better than the basic audit method.

2.2 Secure Cloud Computation Auditing

Definition 1: Cloud Computing Server Threat Model
Suppose that the CS delegates CCS the computation of the function f_1 on the input data set $D = \{m_1, m_2, \ldots, m_n\}$ and each input data m_i is uploaded to the server with position p_i, for $i = 1, 2, \ldots, n$.

The cloud computing server threat is defined as below.

1. The cloud computing server may not correctly and honestly perform the delegated function on all input data $m_1, m_2, \ldots, m_n$.
2. Instead of correctly performing the function f_1 on some input data, the CCS randomly generates the output for the input data.

Definition 2: Secure Cloud Computation Auditing (SCCA for short) Requirement
Suppose the percentage that the CCS does not perform the delegation function f correctly is p. Let the successful cheat be the event that the CS or VA cannot find out

the CCS's threat. The SCCA requirement is satisfied if Pr[successful cheats] < ε, where ε is an allowable probability.

3 Our Proposed Data Strategy

Our brief scheme for cloud computation delegation and auditing is given in Sect. 3.1. Then our improving strategies are proposed in Sects. 3.2, 3.3 and 3.4, respectively.

3.1 Our Brief Cloud Computation Delegation and Auditing Scheme

To simplify the description, suppose that the delegation computation function is only one function f_1. So the computation delegation task $\boldsymbol{F} = \{f_1\}$. It is easy to expand our description for $\boldsymbol{F} = \{f_1, f_2, \ldots, f_{t'}\}$.

Our scheme consists of three phases: Input data preparation and uploading, cloud computation, and computing result auditing phases. Each phase is described briefly. **錯誤! 找不到參照來源。** shows our notations (Table 1).

Table 1. The notations in our auditing strategies

Notations	Definition
$\boldsymbol{F}$	The delegation task function set F = $\{f_1\}$, where f_1 is some delegated function
m_i	Some message block used as the input for the delegation task function
D	An input data set $\{m_1, m_2, \ldots, m_n\}$
D′	An easy-auditor set
D″	The set of extra data for auditing
D_1	The upload data set
R	A computation result set $\{r_1, r_2, \ldots, r_n\}$ of the input data set D
MHTs	Merkle Hash Trees
$Sig_{CU,D}$	CU's signature on the input data set D using CU's private key
$Sig_{CCS,R}$	CCS's signature on the computation result data set R using CCS's private key
t	The number of elements in the easy auditor set
A	The computation result of the easy auditor set
I_N	The index set consisting of the indexes of the challenges choosing form the normal auditor set
I_E	The index set consisting of the indexes of the challenges choosing form the easy auditor set
Challenge = $I_E \cup I_N$	The union of the I_E and I_N

Input Data Preparation and Uploading Phase

Step 1: CU prepares the input data set D = {m_1, m_2, …, m_n} and the extra data set D'' for auditing for the delegation function f_1, where each input data/extra data has its unique index. Let $D_1 = D \cup D''$ be the uploaded data set. D_1 may need be rearranged.

Step 2: CU generates the signature $Sig_{CU,D1}$ on the digest of the uploaded input data set D_1 using CU's private key, where the digest of D_1 is computed using the MHT.

Step 3: CU uploads CCS the uploaded input data set D_1 and the computation delegation task $\boldsymbol{F}$ = {f_1}.

Step 4: CCS validates the signature $Sig_{CU,D1}$ on the uploaded input data set D_1.

Cloud Computation Phase

Step 1: CCS computes the result set $R = \{r_1, r_2, \ldots, r_{n'}\}$ on the uploaded input data set D_1 and the computation delegation task $\boldsymbol{F}$, where n' is the sum of n and the number of elements in D''.

Step 2: CCS generates the signature $Sig_{CCS,R}$ on the computed result set R using CCS's private key and MHT.

Step 3: CCS sends CU the signature $Sig_{CCS,R}$ and the root digest on the result set R, where the root digest of R is computed using the MHT.

Step 4: CU validates the signature $Sig_{CCS,R}$ on the root digest of the result set R.

Computing Result Auditing Phase

Step 0: CU delegates some VA the auditing tasks by sending the index set of D, the signature $Sig_{CU,D1}$, $Sig_{CCS,R}$, the digest on the result set R, and the computation delegation task $\boldsymbol{F}$.

Step 1: VA randomly chooses the index set Challenge from the index set of D to form the auditor set, and sends the Challenge to CCS.

Step 2: CCS sends VA the input data D' whose index are in the index set Challenge, the corresponding computed results R'. CCS also sends the necessary internal digests in the MHTs for the D and R.

Step 3: VA recalculates the root digest of D_1 using the internal digests and D', and validates the $Sig_{CU,D1}$ using the recomputed root digest of D_1.

Step 4: VA recalculates the digest of R using the internal digests and R', and validates the $Sig_{CCS,R}$ using the recomputed digest of R.

Step 5: VA validates the correctness of R'. If all results in R' are correct, then VAs reports CU the cloud computed results are correct this time.

Next our data strategies are proposed by giving the detail in the above cloud computation delegation and auditing scheme.

3.2 The Off-Line Easy-Auditor Improving Strategy

To provide the enough auditing probability against CCS's cheating, the number of auditors in the computed results auditing phase should be large enough to increase the auditing probability as large as VA wants. While the number of auditors is increased, the computation load of the VA is also increased. The increase of the number of auditors also slows down the VA auditing performance, because VA has to compute the results for the randomly chosen auditors. To improve the VA auditing performance, the concept of off-line easy-auditors is proposed.

The off-line easy-auditors are the chosen input data whose results are pre-computed before the computed result auditing phase. Since the results of those chosen input data are computed, the auditing cost for them is just the comparison without performing the delegated function again.

The advantages of the off-line easy-auditors are given here. Trivially, the off-line computation can improve the auditing performance. Moreover, VA provides larger auditing probability against CCS's cheating by choosing more auditors with the help of the off-line easy-auditors. In the following, our first strategy is described by providing the detail for the necessary steps in the input data preparation and uploading phase and computed result auditing phase, respectively.

Input Data Preparation and Uploading Phase

Step 1.1: CU prepares the input data set $D = \{m_1, m_2, \ldots, m_n\}$ for the delegation function f_1. CU and VA cooperatively randomly choose some indexes to construct the subset $D' = \{m'_1, m'_2, \ldots, m'_t\}$ of D as the easy-auditor set, where t is much less than n. After constructing the set D', the different set $D - D'$ forms a normal auditor set. Here, the value of t is dependent on the VA's computational capacity or the auditing probability VA needs. Let $D_1 = D$.

Step 1.2: CU sends VA the D' with the index set and the function f_1.

Step 1.3: VA uses the function f_1 to compute easy auditing value set $A = \{f_1(m'_1), f_1(m'_2), \ldots, f_1(m'_t)\}$.

Computing Result Auditing Phase

In this phase, only the Steps 1 and 5 are stated in detail. The sub-steps for Step 1 are described below.

Step 1.1: VA randomly chooses an index set IE from the index set of D'.

Step 1.2: VA randomly chooses an index set IN from the index set of $D - D'$. VA prepares the easy auditing subset A' from A according to IE, and sends the set Challenge $= I_E \cup I_N$ to CCS.

The detail of Step 5 by VA is described below.

Step 5.1: Partition the R' into two subsets R'_E and R'_N such that R'_E is the subset of the set of the computed results of D' and R'_N is the subset of the set of the computed results of $D - D'$.

Step 5.2: Validate the $R'E$ by comparing whether the results are the same with the corresponding results in A'. If any result is not same, report CU the cloud computed results are incorrect.

Step 5.3: Computes the computing result set A_N of the in-put data subset according the index sex I_N. Validate the R'_N by comparing whether or not any result in R'_N is different from the one in A_N with the same input data index. If anyone is different, report CU the cloud computed results are incorrect; otherwise, report CU the cloud computed results are correct.

3.3 The Function-Based Improving Strategy

The performance limitation of our off-line easy-auditor improving strategy is VA's computation and memory capacity. To overcome the limitation, our next strategy focuses on the properties of the computation function f_1 to increase the number of auditors. Two examples are used to illustrate this strategy. The first is that the function f_1 is a homomorphism function and the second is that the computing problem is isomorphism.

Example 1: Homomorphism Functions
Suppose that the function f_1 is a homomorphism function with respect to the multiplication operation ×. In other words, $f_1(m_1 \times m_2) = f_1(m_1) \times f_1(m_2)$. For example, the RSA encryption function is a famous one with homomorphism. It is easy to see that $f_1(m_1 \times m_2 \times m_3 \times \ldots \times m_k) = f_1(m_1) \times f_1(m_2) \times f_1(m_3) \times \ldots \times f_1(m_k)$. In the following, the value of k is at most 4 as an example.

Input Data Preparation and Uploading Phase

Step 1.1: CU prepares the input data set $D = \{m_1, m_2, \ldots, m_n\}$ for the delegation function f_1. CU randomly chooses the easy-auditor set D' and partitions D' to construct $D'_{E2} = \left\{(m'_{2,1,1}, m'_{2,1,2}), (m'_{2,2,1}, m'_{2,2,2}), \ldots, (m'_{2,t}{}^{2}{}_{,1}, m'_{2,t}{}^{2}{}_{,2})\right\}$, $D'_{E3} = \left\{(m'_{3,1,1}, m'_{3,1,2}, m'_{3,1,3}), (m'_{3,2,1}, m'_{3,2,2}, m'_{3,2,3}), \ldots, (m'_{3,t}{}^{3}{}_{,1}, m'_{3,}{}^{3}{}_{,2}, m'_{3,t}{}^{3}{}_{,3})\right\}$, and $D'_{E4} = \left\{(m'_{4,1,1}, m'_{4,1,2}, m'_{4,1,3}, m'_{4,1,4}), (m'_{4,2,1}, m'_{4,2,2}, m'_{4,2,3}, m'_{4,2,4}), \ldots, (m'_{4,t}{}^{4}{}_{,1}, m'_{4,t}{}^{4}{}_{,2}, m'_{4,t}{}^{4}{}_{,3}, m'_{4,t}{}^{4}{}_{,4})\}\right.$. The number of easy-auditors is t = 2 $t^2 + 3 \times t^3 + 4 \times t^4$, where t is much less than n. The value of t is dependent on the VA's computational capacity or the auditing probability VA needs. The different set D - D′ forms a normal auditor set.

Step 1.2: CU generates $D_1 = D \times D''$, where $D'' = \left\{m'_{2,1,1} \times m'_{2,1,2}, m'_{2,2,1} \times m'_{2,2,2}, \ldots, m'_{2,t}{}^{2}{}_{,1} \times m'_{2,t}{}^{2}{}_{,2}\} \cup \left\{m'_{3,1,1} \times m'_{3,1,2} \times m'_{3,1,3}, m'_{3,2,1} \times m'_{3,2,2} \times m'_{3,2,3}, \ldots, m'_{3,t}{}^{3}{}_{,1} \times m'_{3,t}{}^{3}{}_{,2} \times m'_{3,t}{}^{3}{}_{,3}\} \cup \left\{m'_{4,1,1} \times m'_{4,1,2} \times m'_{4,1,3} \times m'_{4,1,4}, m'_{4,2,1} \times m'_{4,2,2} \times m'_{4,2,3} \times m'_{4,2,4}, \ldots, m'_{4,t}{}^{4}{}_{,1} \times m'_{4,t}{}^{4}{}_{,2} \times m'_{4,t}{}^{4}{}_{,3} \times m'_{4,t}{}^{4}{}_{,4}\}\right.\right.\right.$. After rearranging D_1, CU obtains the new rearranged set D_1 and assigns each element in D_1 a unique index.

Step 1.3: CU sends VA the set D_1, D'_{E2}, D'_{E3}, and D'_{E4} with the index set and the function f_1. Then let D_1 be the unloaded data set.

Computing Result Auditing Phase

Step 1.1: VA randomly chooses some pairs from D'_{E2} to construct D''_{E2}. Since the pair $(m'_{2,x,1}, m'_{2,x,2})$ in D''_{E2} is chosen, the pair $(m'_{2,x,1}, m'_{2,x,2})$ is replaced with $(m'_{2,x,1}, m'_{2,x,2}, m'_{2,x,1}, m'_{2,x,2})$. Then let I'_{E2} be the set of indexes of the input data in D''_{E2}. Similarly, VA randomly chooses some triples from D'_{E3} to construct D''_{E3}. Since $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3})$ in D''_{E3} is chosen, $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3})$ is replaced with $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3}, m'_{3,x,1} \times m'_{3,x,2} \times m'_{3,x,3})$. Let I'_{E3} be the set of indexes of the input data in D''_{E3}. Similarly, VA randomly constructs D''_{E4} from D'_{E4}. Since $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4})$ in D''_{E4} is chosen, the $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4})$ is replaced with $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4}, m'_{4,x,1} \times m'_{4,x,2} \times m'_{4,x,3} \times m'_{4,x,4})$. Let I'_{E4} be the index set of the input data in D''_{E3}. Finally $I_E = I'_{E2} \cup I'_{E3} \cup I'_{E4}$.

Step 1.2: VA randomly chooses the index set I_N from the index set of $D - D'$. Then sends the Challenge = $I_E \cup I_N$ to CCS.

The detail of Step 5 by VA is described below.

Step 5.1: Partition the R' into the subsets $R'_{E2}, R'_{E3}, R'_{E4}$, and R'_N such that R'_{Ei} is the set of the computed results of D''_{Ei}, for i = 2, 3, 4, and R'_N is the subset of the computing result set for D - D′.

Step 5.2: Validate the R'_{E2} by checking the equation $f_1(m'_{2,x,1} \times m'_{2,x,2}) = f_1(m'_{2,x,1} \times m'_{2,x,2})$ for each $(m'_{2,x,1}, m'_{2,x,2}, m'_{2,x,1} \times m'_{2,x,2})$ in D''_{E2}. If anyone does not hold, some CU the cloud computed results are incorrect.

Step 5.3: Validate R'_{E3} by checking $f_1(m'_{3,x,1} \times m'_{3,x,2} \times m'_{3,x,3}) = f_1(m'_{3,x,1}) \times f_1(m'_{3,x,2}) \times f_1(m'_{3,x,3})$ for each $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3}, m'_{3,x,1} \times m'_{3,x,2} \times m'_{3,x,3})$ in D''_{E3}. If anyone does not hold, some cloud computed results are incorrect.

Step 5.4: Validate R'_{E4} by checking $f_1(m'_{4,x,1} \times m'_{4,x,2} \times m'_{4,x,3} \times m'_{4,x,4}) = f_1(m'_{4,x,1}) \times f_1(m'_{4,x,2}) \times f_1(m'_{4,x,3}) \times f_1(m'_{4,x,4})$ for each $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4}, m'_{4,x,1} \times m'_{4,x,2} \times m'_{4,x,3} \times m'_{4,x,4})$ in D''_{E4}. If anyone does not hold, some cloud computed results are incorrect.

Step 5.5: Compute the computed result set R_N of the input data subset according the index set I_N. VA validates the R'_N by comparing whether all the results in R'_N are the same as the ones in R_N with the same index. If anyone is not same, report CU the cloud computed results are incorrect; otherwise, repost CU the cloud computed results are correct.

Example 2: Isomorphism Problem

Suppose that the input of f_1 is a pair (m_{x1}, m_{x2}). The function f_1 is isomorphic if there exist some one-to-one and onto transformations g_x and g'_x such that $f_1(m_{x1}, m_{x2}) = f_1(g_x(m_{x1}), m_{x2}) = f_1(m_{x1}, g'_x(m_{x2})) = f_1(g_x(m_{x1}), g'_x(m_{x2}))$. For example, the function f_1 is to determine whether or not two graphs m_{x1} and m_{x2} are isomorphic.

Input Data Preparation and Uploading Phase

Step 1.1: CU prepares the input data set D = {(m_{11}, m_{12}), (m_{21}, m_{22}), …, (m_{n1}, m_{n2})} for the delegation isomorphic function f_1. CU randomly constructs the subset $D_E = \{(m'_{11}, m'_{12}), (m'_{21}, m'_{22}), \ldots, (m'_{t1}, m'_{t2})\}$ of D and computes $D'' = \{(g_1(m'_{11}), g'_1(m'_{12})), (g_2(m'_{21}), g'_2(m'_{22})), \ldots, (g_t(m'_{t1}), g'_t(m'_{t2}))\}$, where g_x and g'_x are one-to-one and on to functions. Then the uploading input data set is $D_1 = D \cup D''$. Here, the value of t is dependent on the VA's computational capacity or the auditing probability VA needs. The different set D-D_E forms a normal auditor set.

Step 1.2: CU rearranges the set D_1 and assigns each element in D_1 a unique index.

Step 1.3: CU sends VA the index sets of D_1, D_E, and D'' and the function f_1.

Computing Result Auditing Phase

Step 1.1: VA randomly chooses some input data from D_E to construct D'_E. Since the pair (m'_{x1}, m'_{x2}) in D'_E is chosen, the pair $(g_x(m'_{x1}), g'_x(m'_{x2}))$ is replaced with $((m'_{x1}, m'_{x2}),\ (g_x(m'_{x1}), g'_x(m'_{x2})))$. For each $((m'_{x1}, m'_{x2}),\ (g_x(m'_{x1}), g'_x(m'_{x2})))$, those index of m'_{x1}, m'_{x2}, $g_x(m'_{x1})$, and $g'_x(m'_{x2})$ are added into I_E. Finally, I_E is the set of indexes of the input data in D'_E.

Step 1.2: VA randomly chooses the index set I_N from the index set of D-D_E. Then VA sends the set Challenge = $I_E \cup I_N$ to CCS.

The detail of Step 5 by VA is described below.

Step 5.1: Partition the R' into the subsets R'_E and R'_N such that R'_E is the set of the computed results of D'_E and D'_N is the subset of the set of the computed results of D-D_E.

Step 5.2: Validate R'_E by checking whether or not $f_1(m_{x1}, m_{x2}) = f_1(g_x(m_{x1}), g'_x(m_{x2}))$ for each $((m'_{x1}, m'_{x2}),\ (g_x(m_{x1}), g'_x(m_{x2})))$ in D'_E. If anyone does not hold, inform CU the cloud computed results are incorrect.

Step 5.3: Compute the computed result set A_N of the input data subset according to the index sex I_N. Validate the R'_N by comparing whether all the results in R'_N are the same as the ones in A_N with the same input data index. If anyone is not same, report CU the cloud computed results are incorrect; otherwise, repost CU the cloud computed results are correct.

The most on-line auditing load is Step 5.5 in Example 1 and Step 5.3 in Example 2. The number of auditors chosen from the normal auditor set should be small enough that VA can perform it. Moreover, VA also utilizes the computation services of CCS to speed up the auditing. This may leak some information to find out what the easy-auditors are. To overcome this disadvantage, our mixed strategy is proposed.

3.4 The Mixed Improving Strategy

Example 3: Input Data Preparation and Uploading Phase
The sub-steps for the Step 1 by CU is stated

Step 1.1: Prepare the input data set $D = \{m_1, m_2, \ldots, m_n\}$ for the computation function f_1. CU randomly constructs the disjoint subsets $D' = \{m'_1, m'_2, \ldots, m'_t\}$ and $D_E = \{m'_{t+1}, m'_{t+2}, \ldots, m'_{t'}\}$ of D, where t′ is much less than n. Then the different set $D - D' - D_E$ forms the normal auditor set.

Step 1.2: Partition D_E to construct those sets $D'_{E2A} = \{(m'_{2,1,1}, m'_{2,1,2}), (m'_{2,2,1}, m'_{2,2,2}), \ldots, (m'_{2,i}{}^{2}{}_{,1}, m'_{2,i}{}^{2}{}_{,2})\}$, $D'_{E2B} = \{(m'_{2,i}{}^{2}{}_{+1,1}, m'_{2,i}{}^{2}{}_{+1,2}), (m'_{2,i}{}^{2}{}_{+2,1}, m'_{2,i}{}^{2}{}_{+2,2}), \ldots, (m'_{2,t'}{}^{2}{}_{,1}, m'_{2,t'}{}^{2}{}_{,2})\}$, $D'_{E3A} = \{(m'_{3,1,1}, m'_{3,1,2}, m'_{3,1,3}), (m'_{3,2,1}, m'_{3,2,2}, m'_{3,2,3}), \ldots, (m'_{3,i}{}^{3}{}_{,1}, m'_{3,i}{}^{3}{}_{,2}, m'_{3,i}{}^{3}{}_{,3})\}$, $D'_{E3B} = \{(m'_{3,i}{}^{3}{}_{+1,1}, m'_{3,i}{}^{3}{}_{+1,2}, m'_{3,i}{}^{3}{}_{+1,3}), \ldots, (m'_{3,t'}{}^{3}{}_{,1}, m'_{3,t'}{}^{3}{}_{,2}, m'_{3,t'}{}^{3}{}_{,3})\}$, $D'_{E4A} = \{(m'_{4,1,1}, m'_{4,1,2}, m'_{4,1,3}, m'_{4,1,4}), (m'_{4,2,1}, m'_{4,2,2}, m'_{4,2,3}, m'_{4,2,4}), \ldots, (m'_{4,i}{}^{4}{}_{,1}, m'_{4,i}{}^{4}{}_{,2}, m'_{4,i}{}^{4}{}_{,3}, m'_{4,i}{}^{4}{}_{,4})\}$, and $D'_{E4B} = \{(m'_{4,i}{}^{4}{}_{+1,1}, m'_{4,i}{}^{4}{}_{+1,2}, m'_{4,i}{}^{4}{}_{+1,3}, m'_{4,i}{}^{4}{}_{+1,4}), \ldots, (m'_{4,t'}{}^{4}{}_{,1}, m'_{4,t'}{}^{4}{}_{,2}, m'_{4,t'}{}^{4}{}_{,3}, m'_{4,t'}{}^{4}{}_{,4})\}$. Let $D_{EA} = \{D'_{E2A}, D'_{E3A}, D'_{E4A}\}$, $D_{EB} = \{D'_{E2B}, D'_{E3B}, D'_{E4B}\}$. Here, the sizes of $D'_{E2A}, D'_{E3A}, D'_{E4A}, D'_{E2B}, D'_{E3B}, D'_{E4B}$ is dependent on the VA's computational capacity or the auditing probability VA needs.

Step 1.3: Compute $A = \{f_1(m'_1), f_1(m'_2), \ldots, f_1(m'_t)\}$ and $A_{E2A} = \{f_1(m'_{2,1,1} \times m'_{2,1,2}), f_1(m'_{2,2,1} \times m'_{2,2,2}), \ldots, f_1(m'_{2,i}{}^{2}{}_{,1} \times m'_{2,i}{}^{2}{}_{,2})\}$, $A_{E3A} = \{f_1(m'_{3,1,1} \times m'_{3,1,2} \times m'_{3,1,3}), \ldots, f_1(m'_{3,i}{}^{3}{}_{,1} \times m'_{3,i}{}^{3}{}_{,2} \times m'_{3,i}{}^{3}{}_{,3})\}$, $A_{E4A} = \{f_1(m'_{4,1,1} \times m'_{4,1,2} \times m'_{4,1,3} \times m'_{4,1,4}), \ldots, (m'_{4,i}{}^{4}{}_{,1} \times m'_{4,i}{}^{4}{}_{,2} \times m'_{4,i}{}^{4}{}_{,3} \times m'_{4,i}{}^{4}{}_{,4})\}$ and forms the easy auditor set $A_{EA} = \{A_{E2A}, A_{E3A}, A_{E4A}\}$ and A.

Step 1.4: Generate the uploading set $D_1 = D \cup D''$, where $D'' = \{m'_{2,i}{}^{2}{}_{+1,1} \times m'_{2,i}{}^{2}{}_{+1,2}, m'_{2,i}{}^{2}{}_{+2,1} \times m'_{2,i}{}^{2}{}_{+2,2}, \ldots, m'_{2,t2,1} \times m'_{2,t2,2}\} \cup \{m'_{3,i}{}^{3}{}_{+1,1} \times m'_{3,i}{}^{3}{}_{+1,2} \times m'_{3,i}{}^{3}{}_{+1,3}, m'_{3,i}{}^{3}{}_{+2,1} \times m'_{3,i}{}^{3}{}_{+2,2} \times m'_{3,i}{}^{3}{}_{+2,3}, \ldots, m'_{3,t3,1} \times m'_{3,t3,2} \times m'_{3,t3,3}\} \cup \{m'_{4,i}{}^{4}{}_{+1,1} \times m'_{4,i}{}^{4}{}_{+1,2} \times m'_{4,i}{}^{4}{}_{+1,3} \times m'_{4,i}{}^{4}{}_{+1,4}, m'_{4,i}{}^{4}{}_{+2,1} \times m'_{4,i}{}^{4}{}_{+2,2} \times m'_{4,i}{}^{4}{}_{+2,3} \times m'_{4,i}{}^{4}{}_{+2,4}, \ldots, m'_{4,t4,1} \times m'_{4,t4,2} \times m'_{4,t4,3} \times m'_{4,t4,4}\}$. Then rearrange D_1 to obtain the uploaded input data set D_1 and assigns each element in D_1 a unique index.

Step 1.5: Send VA the set $\{D_1, D', D_{EA}, D_{EB}, A, A_{EA}\}$ with the index sets and the function f_1.

Computing Result Auditing Phase

The substeps for Step 1 by VA are described below.

Step 1.1: Randomly choose the index subset $I_{D'}$ from the index set of D'.

Step 1.2: Randomly choose some pairs from D'_{E2A} to construct D''_{E2A}. Since the pair $(m'_{2,x,1}, m'_{2,x,2})$ in D''_{E2A} is chosen, $(m'_{2,x,1}, m'_{2,x,2})$ is replaced with the triple $(m'_{2,x,1}, m'_{2,x,2}, m'_{2,x,1} \times m'_{2,x,2})$. Then let I'_{E2A} be the index set of the elements used in D''_{E2A}. Similarly, VA randomly chooses some triples from D'_{E3A} to construct D''_{E3A}. Since $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3})$ in D''_{E3A} is chosen, $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3})$ is replaced with $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3}, m'_{3,x,1} \times m'_{3,x,2} \times m'_{3,x,3})$. Let I'_{E3A} be the index set of the input data used in D''_{E3A}. Similarly, randomly choose some from D'_{E4A} to construct D''_{E4A}. Since $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4})$ in D''_{E4A} is chosen, the $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4})$ is replaced with $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4}, m'_{4,x,1} \times m'_{4,x,2} \times m'_{4,x,3} \times m'_{4,x,4})$. Let I'_{E2A} be the index set of the input data used in D''_{E4A}. Similarly, D''_{E2B} from D'_{E2B}, D''_{E3B} from D'_{E3B}, and D''_{E4B} from D'_{E4B} are randomly constructed as the auditor sets. Let I'_{E2B}, I'_{E3B}, and I'_{E4B} be the index set of the input data used in D''_{E2B}, D''_{E3B}, and D''_{E4B}, respectively. Finally $D'_E = \{D''_{E2A}, D''_{E3A}, D''_{E4A}, D''_{E2B}, D''_{E3B}, D''_{E4B}\}$ and $I_E = I'_{E2A} \cup I'_{E3A} \cup I'_{E4A} \cup I'_{E2B} \cup I'_{E3B} \cup I'_{E4B}$.

Step 1.3: Randomly chooses the index set I_N from the index set of $D - D' - D''$. Then send the set Challenge = $I_E \cup I_N \cup I_{D'}$ to CCS.

The detail of Step 5 by VA is described below.

Step 5.1: Partition the R' to construct those sets $R'_{E2A}, R'_{E3A}, R'_{E4A}, R'_{E2B}, R'_{E3B}, R'_{E4B}$, $R_{D'}$ and R'_N such that R'_{EiA} and R'_{EiB} are the sets of the computed results of D''_{EiA} and D''_{EiB}, respectively, for i = 2, 3, 4, $R_{D'}$ is the subset of the computed result set with inputs in D', and R'_N is the subset of the computed result set with input data in $D - D' - D_E$.

Step 5.2: Validate $R_{D'}$ by comparing the computed results with the same indexes in $R_{D'}$ and A. VA reports CU that the cloud computed results are incorrect, if any comparison is not the same. Similarly, validate R'_{E2A} with A_{E2A}, R'_{E3A} with A_{E3A}, and R'_{E4A} with A_{E4A}, by comparing whether or not the results with same index are the same. If any comparison is the same, report CU that the cloud computed results are incorrect.

Step 5.3: Validate R'_{E2B} by checking $f_1(m'_{2,x,1} \times m'_{2,x,2}) = f_1(m'_{2,x,1}) \times f_1(m'_{3,x,2})$ for each $(m'_{2,x,1}, m'_{2,x,2}, m'_{2,x,1} \times m'_{2,x,2})$ in D''_{E2B}. If anyone is different, tell CU the cloud computed results are incorrect.

Step 5.4: Validate R'_{E3B} by checking $f_1(m'_{3,x,1} \times m'_{3,x,2} \times m'_{3,x,3}) = f_1(m'_{3,x,1}) \times f_1(m'_{3,x,2}) \times f_1(m'_{3,x,3})$ for each $(m'_{3,x,1}, m'_{3,x,2}, m'_{3,x,3}, m'_{3,x,1} \times m'_{3,x,2} \times m'_{3,x,3})$ in D''_{E3B}. If anyone is different, some cloud computed results are incorrect.

Step 5.5: Validate R'_{E4B} by $f_1(m'_{4,x,1} \times m'_{4,x,2} \times m'_{4,x,3} \times m'_{4,x,4}) = f_1(m'_{4,x,1}) \times f_1(m'_{4,x,2}) \times f_1(m'_{4,x,3}) \times f_1(m'_{4,x,4})$ for each $(m'_{4,x,1}, m'_{4,x,2}, m'_{4,x,3}, m'_{4,x,4}, m'_{4,x,1} \times m'_{4,x,2} \times m'_{4,x,3} \times m'_{4,x,4})$ in D''_{E4B}. If anyone is different, some cloud computed results are incorrect.

Step 5.6: Compute the computed result set A_N with the input data subset according to the index set I_N. Validate the R'_N by finding out whether or not any results in R'_N is different from the corresponding one in A_N with the same input data index. If any difference is found, report CU the cloud computed results are incorrect; otherwise report CU the cloud computed results are correct.

4 Conclusions

To improve the on-line audit performance, three data strategies, the off-line easy-auditor improving strategy, the function-based improving strategy and mix improving strategy are proposed. According to the concept of off-line easy-auditor set and off-line computation, the online performance of audit is improved. Due to the computation limitation, this strategy is bounded by either the delegated function hardness and auditors' computation and storage. To remove the limitation, in our function-based improving strategy, auditors utilize the cloud computation server to compute the results for the auditors. Please note that, if your email address is given in your paper, it will also be included in the meta data of the online version.

References

1. Wang, B., Li, B., Li, H.: Knox: privacy-preserving auditing for shared data with large groups in the cloud. In: Applied Cryptography and Network Security, ACNS 2012. LNCS, vol. 7341, pp. 507–525. Springer, Heidelberg (2012)
2. Wang, B., Li, B., Li, H.: Oruta: Privacy-Preserving Auditing for Shared Data in the Cloud. In: Proceeding of IEEE 5th International Conference on Cloud Computing, IEEE Cloud 2012, pp. 295–302, Honolulu, HI, U.S.A, 24–29 June 2012
3. Wang, B., Li, B., Li, H.: Privacy-preserving public auditing for shared cloud data supporting group dynamics. In: Proceeding of IEEE International Conference on Communications, ICC 2013, pp. 1946–1950, Budapest, Hungary, 9–13 June 2013
4. Wang, C., Ren, K., Lou, W., Li, J.: Toward public auditable secure cloud data storage services. IEEE Netw. **24**(4), 19–24 (2010)
5. Wang, Q., Wang, C., Ren, K., Lou, W., Li, J.: Enabling public auditability and data dynamics for storage security in cloud computing. In: Proceeding of ESORICS 2009, Saint Malo, France, 21–25 September 2009, pp. 355–370 (2009)
6. Belenkiy, M., Chase, M., Erway, C., Jannotti, J., Küpçü, A., Lysyanskaya, A.: Incentivizing outsourced computation. In: Proceedings of the 3rd International Workshop on Economics of Networked Systems, pp. 85–90, Seattle, WA, U.S.A, 17–22 August 2008
7. Canetti, R., Riva, B., Rothblum, G.: Verifiable computation with two or more clouds. In: Workshop on Cryptography and Security in Clouds, Zurich, Switzerland, 15–16 March 2011

8. Gennaro, R., Gentry, C., Parno, B.: Non-interactive verifiable computing: outsourcing computation to untrusted workers. In: 30th International Cryptology Conference, CYPTO 2010, pp. 465–482, Santa Barbara, California, U.S.A, 15–19 August 2010
9. Golle, P., Mironov, I.: Uncheatable distributed computations. In: The Cryptographers' Track at RSA Conference 2001, pp. 425–440, San Francisco, CA, U.S.A, 8–12 April 2001
10. Sadeghi, A., Schneider, T., Winandy, M.: Token-based cloud computing: secure outsourcing of data and arbitrary computations with lower latency. In: Trust and Trustworthy Computing, pp. 417–429, Berlin, Germany, 21–23 June 2010
11. Wang, Q., Wang, C., Ren, K., Lou, W., Li, J.: Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Trans. Parallel Distrib. Syst. **22**(5), 847–859 (2012)
12. Wei, L., Zhu, H., Cao, Z., Jia, W., Vasilakos, A.: Seccloud: bridging secure storage and computation in cloud. In: 30th International Conference on Distributed Computing Systems Workshops, IEEE ICDCSW 2010, Genova, Italy, 21–25 June 2010
13. Wei, L., Zhu, H., Cao, Z., Jia, W., Dong, X., Jia, W., Chen, Y., Vasilakos, A.: Security and privacy for storage and computation in cloud computing. Inf. Sci. **258**, 371–386 (2014)
14. Monrose, F., Wycko, P., Rubin, A.: Distributed execution with remote audit. In: Proceedings of the Network and Distributed Systems Security Symposium (NDSS), San Diego, California, U.S.A, pp. 103–113 (1999)

Information Hiding and Secret Sharing

Reversible Image Steganography for Color Image Quantization Based on Lossless Index Coding

Yu-Chen Hu[1](✉), Chin-Feng Lee[2], and Yi-Hung Liu[3]

[1] Department of Computer Science and Information Management, Providence University, Taichung, Taiwan
ychu@pu.edu.tw

[2] Department of Information Management, Chaoyang University of Technology, Taichung, Taiwan
lcf@cyut.edu.tw

[3] Business School, Shantou University, Guangdong, China
lyh0315@gmail.com

Abstract. In this paper, we proposed a joint lossless index coding and data hiding technique for the palette images. The palette image is the compressed image of the color image quantization technique. The compressed codes of the palette image consist of the index table and the color palette. In the proposed technique, a three-category lossless index coding method is employed. The secret data is embedded into the encoded index table during the index coding process is executed. From the results, it is shown that good hiding capacity is obtained in the proposed technique while keeping a good bit rate.

Keywords: Reversible data hiding · Color image quantization
Lossless index coding · Palette design

1 Introduction

Image steganography is one of the major categories of information hiding, which conceals secret data into a digital image [1]. The common requirements in image steganography are achieving high hiding capacity, providing good visual imperceptibility, and ensuring security against steganalysis [2, 3].

In general, image steganography can be further classified into two main approaches: irreversible [4–6], reversible [7–12]. In the irreversible approach, the cover image is modified to embed secret data and cause the distortions to the image permanently. In other words, the embedded image cannot be reversed to its original image. The irreversible approach is suitable for the general-purposed images.

In the reversible approach, the cover image is modified to embed secret data. The embedded secret data can be extracted from the embedded image and the original image can also be exactly recovered. It is particularly suitable for important images such as military images, medical images, and satellite maps.

The reversible image steganography schemes are also called the lossless image steganography schemes. Basically, two main categories of the reversible image

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 185–195, 2018.
https://doi.org/10.1007/978-3-319-76451-1_18

steganography schemes had been introduced. They are difference expansion [9] and histogram shifting [10–12]. Some reversible data hiding schemes that use the mix of these two approaches had also been proposed.

The basic concept of difference expansion had been introduced in [9]. In this scheme, each pair of the two neighboring pixels is used to embed at most 1-bit secret data. The difference and average value of two neighboring pixels in each pair are calculated. The secret data to be embedded is appended to the difference value represented as a binary number. Then the difference value and 1-bit secret data are added. The calculated result is stored in the modified difference value. Two modified neighboring pixels are replaced by the summation and subtraction of the average value.

The concept of the reversible image steganography based on histogram shifting had been proposed in [10]. In this scheme, the image histogram of the pixels in the cover image is produced. The required pairs of the peak points and the zero points are searched. Each pixel in the peak point is used to embed 1-bit secret data. The others are modified and no secret data are embedded. In this scheme, the summation of the number of pixels in the peak points is the maximal hiding capacity for the secret data to be embedded.

To increase the hiding capacity of the histogram shifting, a block-based image steganography scheme based on residual histogram shifting had been proposed [11]. In this scheme, the cover image is divided into non-overlapped image blocks. Each pixel in the block is processed by using the linear prediction technique to generate the prediction errors. The residual histogram of the cover image is employed to embed the secret data. In addition, the reversible data hiding scheme based on histogram shifting of n-bit planes had been proposed [12].

In previous studies, most image steganography techniques work on the digital image in raw format. Some image steganography techniques for the compressed images of vector quantization, block truncation coding, sub-band coding, color image quantization, JPEG, JPEG 2000 had also been proposed. Typically, the hiding capacity of one image steganography technique for the raw image is higher than that of one image steganography technique for the compressed image.

In this study, we designed a reversible image steganography for the compressed image of color image quantization (CIQ) [13–16]. CIQ is a commonly used image coding technique for color images. To cut down the storage cost of one RGB color image, first of all, the color palette of k representative color pixels is generated. Then, the closest color pixel for each color pixel in the RGB color image is searched and its corresponding index is recorded. The compressed result of one RGB color image consists of the set of indices and the color palette used. The compressed image of CIQ is often called the palette image.

The study of the lossless index coding technique for color image quantization [16] motivates us the design of the proposed technique. The high degree of similarity among neighboring indices is exploited in this technique. To increase the degree of similarity among neighboring indices, the color palette used in the pixel mapping is sorted previously. The indices that were compressed by CIQ are classified into three categories. Different encoding rule has been adopted to encode one index in every category. Experimental results show that the proposed technique significantly cuts down the number of bit rates without incurring extra image degradation.

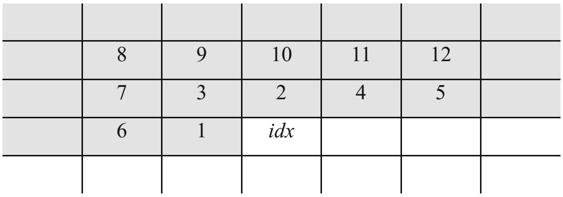
2 The Proposed Technique

The goal of the proposed technique is to encode the indices of CIQ and embed secret data altogether. The color palette of k colors used in CIQ is first sorted by the mean values of the color pixels. By doing so, the similar color pixels have a higher possibility to be arranged in the neighboring area of the color palette. In the lossless index coding process, each index will be classified into one of the three categories. Only the indices belong to the first two categories will be used to embed secret data. No secret data will be embedded into the indices of the third category.

The input dataset is the indices of the compressed image by CIQ. The set of indices is often called the index table. Suppose the color image of $W \times H$ pixels is compressed by CIQ using the sorted color palette of k colors. The index table consists of $W \times H$ indices of $\log_2 k$ bits.

Each index *idx* in the index table will be classified as one of the three categories. Let *SNO* denote the number of the distinct indices to be searched in the neighboring area. Among these *SNO* entries, one entry is reserved as a switch code for data embedding. In other words, only *SNO*-1 distinct neighboring indices will be checked.

If the same index of *idx* can be found among these *SNO*-1 distinct neighboring indices, *idx* is classified as the first category. The position code of its identical index is stored in $\log_2$*SNO* bits. The search order of the indices in the neighborhood for finding the same index is depicted in Fig. 1. Only the non-repeated indices are used to encode the current index *idx*, because it is highly possible that the same indices are appeared in the neighboring area.



	8	9	10	11	12	
	7	3	2	4	5	
	6	1	*idx*			

Fig. 1. Search order of the encoded neighbors for the current index *idx*

An illustrative example for the index coding of the first category is depicted in Fig. 2. Suppose *SNO* is set to 4 in this example. Four distinct indices 114, 113, 115, and 116 are searched at positions 1, 2, and 4. Here, the last entry of *SNO* is reserved. We discover that the third distinct index has the same value as the current index 115. Here, 2-bit position code $(10)_2$ is stored and used to encode the current index.

	116	114	113	113	115	
	116	114	**113**	**115**	115	
	116	**114**	115			

Fig. 2. Example of the index encoding of the first category

To continue the same example as shown in Fig. 2, the same index cannot be found in the neighboring areas of the current index 115 when *SNO* is set to 2. It is obvious that the *SNO* value has great influence on the percentage of the indices belonging to the first category. The larger *SNO* value is set, the higher possibility it is that the same index can be found in the neighboring area of the current index.

If the current index *idx* does not belong to the first category, the difference value (*dv*) between *idx* and the reference index *ri* is computed as follows

$$dv = idx - ri. \tag{1}$$

Here, *ri* is set to the adjacent left index of *idx*. Let *RTH* denote the tolerant range threshold for relatively address.

After calculating *dv*, *idx* will be classified as the second category if *dv* is no less than –*RTH*/2 and no greater than (*RTH*/2) – 1. Otherwise, *idx* will be classified as the third category. For each index belonging to the third category, the original index of $\log_2 k$ bits is stored.

	dv
0	-*RTH*/2
1	-*RTH*/2+1
2	-*RTH*/2+2
	⋮
⋮	-2
	-1
	1
	2
	⋮
	RTH/2-1
RTH-1	*RTH*/2

(a) Possible difference values

	diff	codes
0	-4	$(000)_2$
1	-3	$(001)_2$
2	-2	$(010)_2$
3	-1	$(011)_2$
4	1	$(100)_2$
5	2	$(101)_2$
6	3	$(110)_2$
7	4	$(111)_2$

(b) Example with *RTH* = 8

Fig. 3. Encoding example of the relatively addressing approach

To encode the index that belongs to the second category, the relative addressing of the difference value is employed. Figure 3(a) shows the possible difference values –*RTH*/2, –*RTH*/2 + 1, …, –2, –1, 1, 2, …, *RTH*/2 – 1 of the second category. There are *RTH*-1 difference values for the relative addressing. Additional one entry with value *RTH*/2 is reserved for data embedding.

These difference values are encoded in $\log_2(RTH)$ bits in the relative addressing process. The decimal values of their codes range from 0 to *RTH*-1. A coding example of the difference values when *RTH* is set to 8 is depicted in Fig. 3(b). In this example, there are 8 possible different values. The decimal values of 3 bits range from 0 to 7.

To embed 1-bit secret data *sd* in one index of the first category, the resultant code is the position code of $\log_2 SNO$ bits if *sd* equals 0. Otherwise, the switch code with value *SNO*-1 and the position code are stored.

To embed 1-bit secret data *sd* in one index of the second category, the resultant code is the difference code for relative addressing if *sd* equals 0. Otherwise, the switch code with value *RTH*-1 and the relative addressing code are stored. The details of the code patterns and the code lengths of the index coding and the data embedding process mentioned above are listed in Table 1.

Table 1. Code pattern and the code length of the index coding and the data embedding process

Factors		Code pattern	Code length
Category 1	$sd = 0$	position code	$\log_2 SNO$
	$sd = 1$	switch + position code	$2 \times \log_2 SNO$
Category 2	$sd = 0$	difference code	$\log_2 RTH$
	$sd = 1$	switch + difference code	$2 \times \log_2 RTH$
Category 3	N/A	index	$\log_2 k$

In addition to the resultant code for each index of each category, the indicators for these three categories are needed. The indicator of each category is placed in front of the compressed codes of the index in each category so that each index can be correctly decoded.

Three indicators $indr_1$, $indr_2$, and $indr_3$ will be designed by using the Huffman coding technique based on the occurrences of the indices in these three categories. Let no_1, no_2, and no_3 denote the total number of indices of these three categories, respectively. The indicator corresponding to the largest occurrence is encoded using one bit. The others are encoded using two bits.

An example of the indicator generation based on entropy coding is depicted in Fig. 4. Suppose no_1, no_2, and no_3 equal 6000, 3000, and 1000, respectively. Three nodes N_1, N_2, and N_3 that correspond to these three categories are produced. First, N_3 and N_2 are selected and merged into T_1. Then the merged node T_1 and N_1 are then merged together into the root node T_2. In this example, the resultant codes of these three indicators $indr_1$, $indr_2$, and $indr_3$ are $(1)_2$, $(01)_2$, and $(00)_2$, respectively.

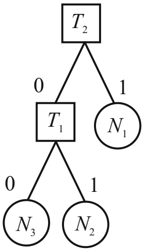


Fig. 4. Example of indicator generation based on entropy coding

3 Simulation Results

In the simulations, six color images of 512 × 512 pixels "Airplane", "Lenna" "Pepper", "Sailboat", "Splash" and "Tiffany" as shown in Fig. 5 are used as the testing images for performance comparison. The color palettes are designed by using the fast splitting algorithm [15].

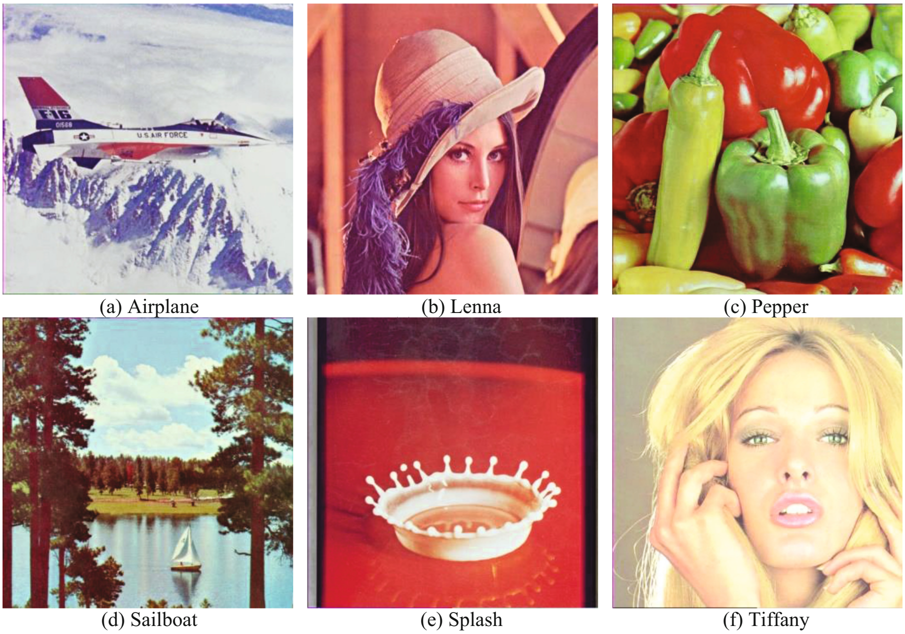

(a) Airplane (b) Lenna (c) Pepper
(d) Sailboat (e) Splash (f) Tiffany

Fig. 5. Six test images of 512 × 512 pixels

To measure the image quality of the compressed image, the Mean Square Error (*MSE*) between the pixels of the original image and those of the enlarged image is defined as:

$$MSE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} (o_{ij} - e_{ij})^2. \tag{2}$$

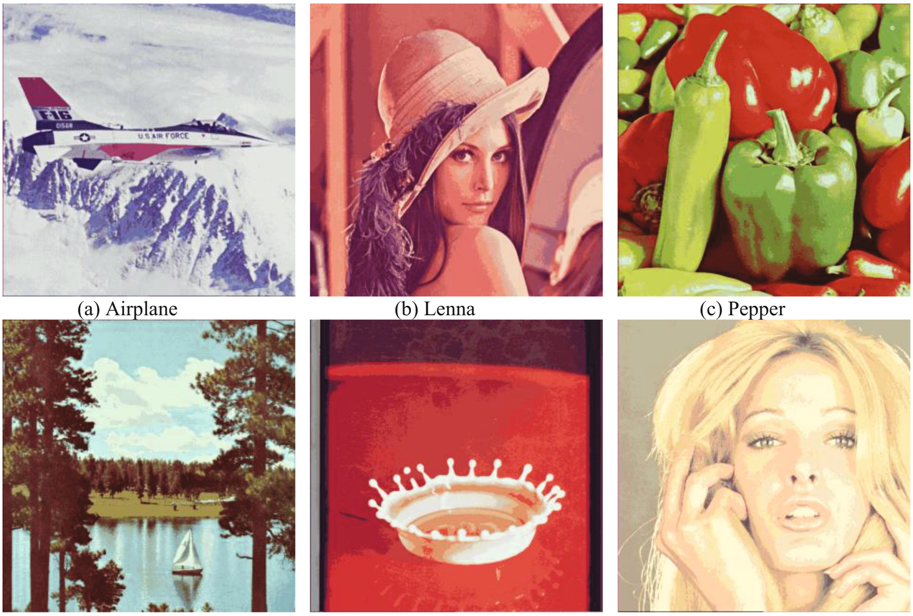
Here, o_{ij} and e_{ij} denote the color pixels in the original color image and the compressed color image, respectively.

The quality of the palette image is measured by means of the peak signal-to-noise-ratio (*PSNR*), which is defined as

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}. \tag{3}$$

Basically, *PSNR* is considered as an indication of image quality rather than a definitive measurement; however, it is a commonly used measurement for evaluating the image quality.

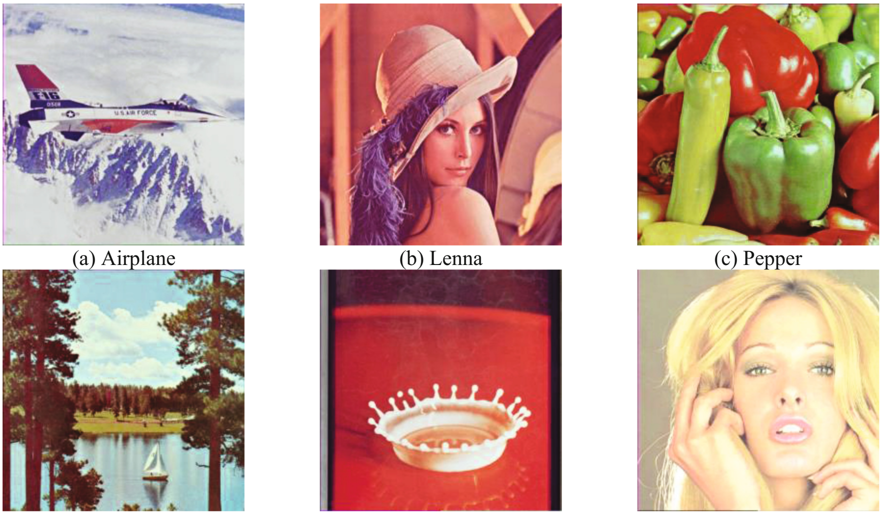
Reconstructed images of CIQ using color palettes of size 16 and 256 are shown in Figs. 6 and 7, respectively. It is shown that the false contour effect can be easily found in the reconstructed images when the color palettes of size are used. The visual qualities of these images are bad. When the color palettes of size 256 are used, good reconstructed image qualities of the CIQ compressed images are obtained. The required bit rates of CIQ using the color palettes of size 16 and 256 equal 4 bpp and 8 bpp, respectively.



(a) Airplane (b) Lenna (c) Pepper

(d) Sailboat (e) Splash (f) Tiffany

Fig. 6. Compressed images of CIQ using color palettes of 16 colors



(a) Airplane (b) Lenna (c) Pepper

(d) Sailboat (e) Splash (f) Tiffany

Fig. 7. Compressed images of CIQ using color palettes of 256 colors

As shown in Table 2, with the increasing of the palette size, the image quality grows in the color image quantization technique. Average image qualities of 28.173 dB, 32.629 dB and 37.143 dB are achieved by using the pixel copy technique when the palette sizes are 16, 64 and 256, respectively.

Table 2. Image qualities of the color image quantization technique By Using different palette sizes

Images	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
Airplane	30.438	32.553	35.425	38.079	40.096
Lenna	28.710	31.010	33.533	35.475	37.325
Pepper	26.205	28.250	30.055	32.315	34.728
Sailboat	27.068	28.987	30.630	32.422	34.486
Splash	28.365	30.830	33.105	36.414	38.776
Tiffany	28.253	30.709	33.024	35.318	37.447
Average	**28.173**	**30.390**	**32.629**	**35.004**	**37.143**

Experimental results of the bit rates by using the three-category index coding method [16] are listed in Fig. 8. In the simulations, the values of *SNO* are set to 2, 4, and 8. From the results, it is shown that the best bit rates are achieved when *RTH* is set to 32. In addition, the lossless index coding method achieves the best performance when *SNO* is set to 4.

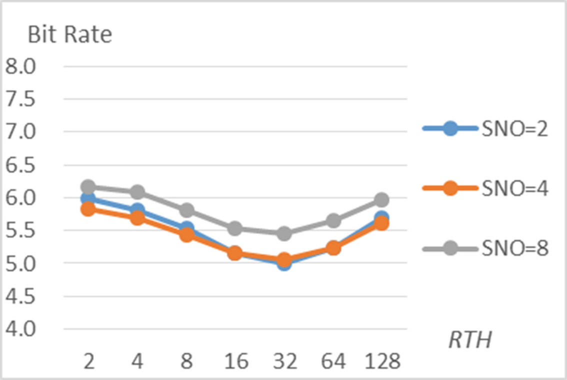


Fig. 8. Average bit rates of the lossless index coding method

Since the design of the proposed technique is based on the study of the lossless index coding method [16], as shown in Fig. 8, the results help us understand the upper bounds of the bit rates of the proposed technique. The lossless index coding process used in the proposed technique reserves one entry as the switch code for the encoding of indices in both the first and the second categories. The numbers of the actually used entries of the comparative method and the proposed technique are listed in Table 3.

Table 3. Analysis of the actually used entries in the comparative method (CM) [16] and the proposed technique (PT) for lossless index coding

Factors	CM [16]	PT
Category 1	*SNO*	*SNO*-1
Category 2	*RTH*	*RTH*-1

Average bit rates and hiding capacities of the proposed technique with different *SNO* values are listed in Figs. 9 and 10, respectively. In Fig. 9, the proposed technique achieves the lowest bit rates when *SNO* is set to 2. The best bit rates of the proposed technique with *SNO* valued 2, 4, 8 are achieved when the values of *RTH* is set to 16, 16, and 32, respectively.

From the results in Fig. 10, it is shown that the hiding capacity of the proposed technique increases as the increment of the *RTH* value. The proposed technique achieves the highest hiding capacities when *SNO* is set to 8.

The selection of the thresholds *SNO* and *RTH* has great influence on the performance of the proposed techniques. The results listed in Figs. 9 and 10 help us to select the controlling threshold values. To embed the secret data of 100000 bits by using the proposed technique, *SNO* and *RTH* are suggested to be set to 2 and 2, respectively. Average bit rate of 6.745 bpp is achieved while keeping an average hiding capacity of 0.409 bit per index (bpi). To embed the secret data of 200000 bits by using the proposed technique, *SNO* and *RTH* are suggested to be set to 4 and 16, respectively. Average bit rate of 6.348 bpp is achieved by using the proposed technique while keeping an average hiding capacity of 0.764 bpi.

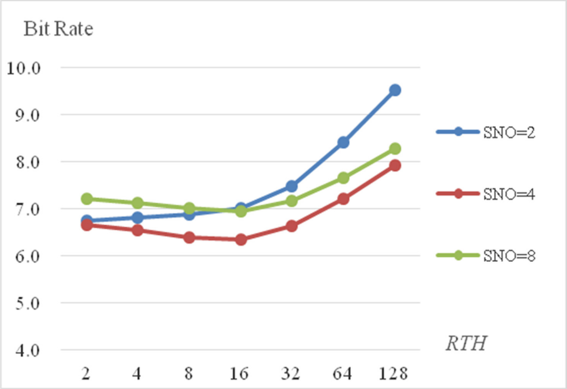


Fig. 9. Average bit rates of the proposed technique with different *SNO* values

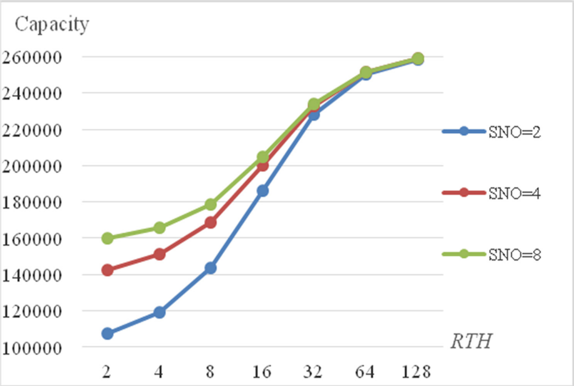


Fig. 10. Average hiding capacities of the proposed technique

4 Conclusions

A joint lossless index coding and data embedding technique had been proposed in this paper. The input of the proposed technique can either be the RBG color image or the palette image. From the results, it is shown that average hiding capacities ranging from 0.409 bpi to 0.988 bpi are obtained by using the proposed technique. From the results, it is suggested that the *SNO* value should be set to 4 in the proposed technique.

According to the results, the bit rates of the proposed technique are greater than the bit rates of CIQ in some cases. Note that 8 bpp is required by using CIQ when the color palette of 256 colors is required. For example, average bit rates of 8.409 bpp and 9.521 bpp are required by using the proposed technique with *SNO* equal 2 when *RTH* values are set to 64 and 128, respectively. In addition, average bit rate of 8.263 bpp is consumed by using the proposed technique with *SNO* equal 8 when *RTH* values is set to 128, respectively. To solve this problem, we will try to improve the proposed technique to further cut down the bit rates while keeping good hiding capacities.

Acknowledgments. This research was supported by the National Science Council, Taipei, R.O.C. under contract MOST 106-2221-E-324-006-MY2, MOST 106-2410-H-126-006-MY2 and MOST 103-2410-H-126-009-MY3.

References

1. Provos, N., Honeyman, P.: Hide and seek: an introduction to steganography. IEEE Secur. Priv. **1**, 32–44 (2003)
2. Cheddad, A., Condell, J., Curran, K., Mc Kevitt, P.: Digital image steganography: survey and analysis of current methods. Sig. Process. **90,** 727–752 (2010)
3. Subhedar, M.S., Mankar, V.H.: Current status and key issues in image steganography: a survey. Comput. Sci. Rev. **13**, 95–113 (2014)
4. Chang, C.C., Lin, M.H., Hu, Y.C.: A fast and secure image hiding scheme based on LSB substitution. Int. J. Pattern Recogn. Artif. Intell. **16**, 399–416 (2002)
5. Swain, G.: Adaptive pixel value differencing steganography using both vertical and horizontal edges. Multimedia Tools Appl. **75**, 1–16 (2015)
6. Hussain, M., Wahab, A.W.A., Ho, A.T.S., Javed, N., Jung, K.H.: A data hiding scheme using parity-bit pixel value differencing and improved rightmost digital replacement. Sig. Process. Image Commun. **50**, 44–57 (2017)
7. Chang, C.C., Yu, Y.H., Hu, Y.C.: Hiding secret data in images via predictive coding. Pattern Recogn. **38**, 691–705 (2005)
8. Qin, C., Hu, Y.C.: Reversible data hiding in VQ index table with lossless coding and adaptive switching mechanism. Sig. Process. **129**, 48–55 (2016)
9. Tian, J.: Reversible data embedding using a difference expansion. IEEE Trans. Circ. Syst. Video Technol. **13**, 890–896 (2003)
10. Ni, Z., Shi, Y.Q., Ansari, N., Su, W.: Reversible data hiding. IEEE Trans. Circ. Syst. Video Technol. **16**, 354–362 (2006)
11. Tsai, P.Y., Hu, Y.C., Yeh, H.L.: Reversible image hiding scheme using predictive coding and histogram shifting. Sig. Process. **89**, 1129–1143 (2009)
12. Liu, L., Chang, C.C., Wang, A.: Reversible data hiding scheme based on histogram shifting of n-bit planes. Multimedia Tools Appl. **75**, 11311–11326 (2016)
13. Michael, T., Charles, A.: Color quantization of images. IEEE Trans. Sig. Process. **39**, 2677–2690 (1991)
14. Hu, Y.C., Li, M.G.: A k-means based color palette design scheme with the use of stable flags. J. Electron. Imaging **16**, p. 1–11 (2007). art no. 033003
15. Hu, Y.C., Li, M.G., Tsai, P.Y.: Color palette generation schemes for color image quantization. Imaging Sci. J. **57**, 46–59 (2009)
16. Hu, Y.C., Chiang, C.Y., Chen, W.L., Chou, W.K.: Lossless Index Coding for Indexed Color Images. Imaging Sci. J. **60**, 54–63 (2012)

Capacity on Demand Steganography Through Adaptive Threshold Strategy

Sheng-Chih Ho[1(✉)], Chung-Yi Lin[1], and Chao-Lung Chou[2]

[1] Department of Information Management, National Defense University, Taipei, Taiwan
hszndu@gmail.com

[2] Department of Computer Science and Information Engineering, National Defense University, Taoyuan, Taiwan

Abstract. In the process of secret communication, in fact, the length of each transmission of ciphertext will never be the same. In order to meet dynamic length of ciphertext and provide a good image quality, a novel adaptive threshold strategy is presented. Firstly, a threshold pixel value and a k value are decided dynamically based on the length of ciphertext and the cumulative statistics of cover-image's pixel value. Then if a pixel value is more than threshold value, more than k bits secret data can be embedded by using modified Least Significant Bit (LSB) substitution method, or the data of equal or less than k bits can be embedded. This dynamic strategy can adjust hiding capacity consistent with the length of secret message, and the secret data can be embedded in stego-image as evenly as possible. Quality of stego-image would be improved by this way. The experimental results, the proposed method not only achieves a larger embedding capacity, but also has higher visual quality of stego-image than most of proposed LSB based methods.

Keywords: Adaptive steganography · Threshold pixel value · Spatial-domain · Modified LSB substitution

1 Introduction

Due to the rapid development of Internet, digital data (such as text, image, audio, video and so on.) can be exchanged quickly by Internet. Cryptography may provide a safe way to protect the sensitive digital content [1]. However, the constructed ciphertext is a meaningless message which will easily attracts illegal users' attention and destruction. To overcome this problem, steganography offers a different approach to transmit secret messages, which can embed secret information into cover-media such that hackers cannot find out the existence of the secret message [2]. In general, the technique of steganography usually focuses on the following three parts [3].

(1) Maximum of data hiding capacity.
(2) Optimization of stego-image quality.
(3) Stegonagrpahic method can be effective against steganalysis.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 196–205, 2018.
https://doi.org/10.1007/978-3-319-76451-1_19

There are four basic spatial domain steganography techniques that have been proposed [4–7]. A well-known and most common steganographic method is directly replacing k least significant bits (LSBs) of pixel value in the cover image with k secret bits [4]. LSB method typically achieves simple, high capacity and less computational. In 2004, Chan and Cheng et al. propose a data hiding scheme by simple LSB substitution with an optimal pixel adjustment process (OPAP) [5] is a modified LSB method and the worst mean-square-error (WMSE) is less than 1/2 of Bender et al.'s method [4]. Thus the quality of stego-image in OPAP scheme is better than that of the LSB substitution scheme.

In 2003, Thien and Lin propose a simple data hiding method for high-hiding capacity based on modulus function [6]. This scheme is almost as simple as the LSB substitution method [4] in both embedding and extracting, and the error of most pixel values between cover-image and stego-image is smaller than $\lceil (m - 1)/2 \rceil$. Therefore, this method also has better quality of stego-image than half of the LSB substitution scheme in the same hiding capacity.

In general, the edge regions of image can tolerate more alteration than smooth regions. However, the LSB-based methods embed the secret message into cover-image directly without considering above concept. In 2003, Wu and Tsai propose a pixel-value differencing (PVD) steganographic method to improve the above disadvantage [7]. The hiding capacity of PVD method is determined by the different value between the pixels, in which two consecutive pixels are formed a sub-block to embed secret data. If the different value of two consecutive pixels is large, that means the sub-block belongs to an edge area, and more secret data can be embedded here. On the other hand, if the different value is small, that means the sub-block belongs to a smooth area, and less secret data can be hidden.

There are many spatial domain steganographic schemes [8–20] have been proposed based on the above mentioned these four basic data hiding methods. This is the reason why these four methods are called the basic methods.

In this paper, the proposed method provides a novel adaptive data hiding strategy to enhance the hiding capacity and stego-image's quality. At first, a threshold pixel value is adjusted adaptively according to the length of secret data stream by the adaptive strategy algorithm. If the pixel value is bigger than the threshold value, more secret data can be embedded here. On the contrary, if the pixel value is equal or smaller than the threshold value, less secret data can be hidden here. In this way, the secret message can be embedded in the cover image evenly. In order to keep the quality of the stego-images, the proposed method will apply the modified LSB substitution method [5] to embed secret data. The concept of modified LSB substitution is to increase or decrease the most-significant-bit (MSB) part by 1 for reducing the square error between the original pixel and the embedded pixel. For instance, secret data is $s = 101_{(2)}$ then a pixel $P = 10111000_{(2)}$ is embedded by the 3-bit simple LSB substitution method and the result $P' = 10111101_{(2)}$. The MSB part of P' is decreased by 1, so the result of modified LSB substitution method is $P' = 10110101_{(2)}$, which largely reduce the error between P and P'.

The rest of this paper is organized as follows. The proposed method is presented in Sect. 2. In Sect. 3, the experimental results and discussions will be shown. Finally, conclusions are given in Sect. 4.

2 Data Hiding by Adaptive Threshold Strategy

The threshold of the proposed adaptive data hiding strategy scheme is produced dynamically based on the hiding capacity requirement (length of ciphertext) and the cumulative statistics of cover-image's pixel value. In general, the hiding capacities are not fixed number and the cover-images have different pixel value distribution in each data hiding task. The proposed method utilizes above two dynamic parameters to obtain the threshold pixel value. The range [0,255] of pixel value will be divided into higher level and lower level by the threshold pixel value. For example, threshold pixel value is 160, the range [0,160] is belonging to lower level and range [161,255] is belonging to higher level. Every pixel is embedded by the k bits or (k + 1) bits modified LSB substitution. If the pixel belongs to higher level and it will be embedded k + 1 bits secret data, otherwise it can only be embedded k bits. In order to extract the embedded data correctly, the same location pixel values of cover image and stego-image should belong to the same level. If the pixel value changes into another level after embedding, an emendation procedure is used to amend this error. The proposed adaptive threshold generation, embedding and extracting algorithms are presented in subsections below.

2.1 Adaptive Threshold Generation Scheme

Step 1: Given a 256 gray-scale cover image *F* with size of $M \times N$. The *S* is the secret data stream that will be embedded into *F*. *CT* is the length of *S*. At first, all pixels of the image *F* are scanned in roster scan order, and the number of occurrences of every pixel value are recorded respectively. Then 256 variables $P_t(0)$, $P_t(1)$, $P_t(2)$,…., and $P_t(255)$ can be obtained, where $P_t(0)$, $P_t(1)$, $P_t(2)$, …., and $P_t(255)$ are the number of occurrences of pixel value 0,1,2,…., and 255 in *F* respectively.

Step 2: c_0 is the embedding capacity of lower level pixel and c_1 is the embedding capacity of higher level pixel. Two variables can be obtained by following rules.

$$\boxed{\begin{array}{l} \textbf{Given}: 1 \le c_0 \le 4, 1 \le c_1 \le 5, \\ \qquad\qquad c_0 \le c_1, CT \le 4.5 \times M \times N \\ \textbf{Case } 1: \dfrac{CT}{M \times N} \le 1, \\ \qquad\qquad c_0 = 1,\ c_1 = 1; \\ \textbf{Case } 2: \dfrac{CT}{M \times N} > 1\,, \\ \qquad\qquad c_0 = \left\lfloor \dfrac{CT}{M \times N} \right\rfloor, c_1 = \left\lceil \dfrac{CT}{M \times N} \right\rceil; \end{array}} \tag{1}$$

Step 3: The threshold pixel value *T* can be derived by following formula.

$$\boxed{\begin{array}{l} \textbf{Given}: 0 \leq j < 255 \\ \left[\sum_{i=0}^{j} \mathrm{P_t(i)} \times c_0 + \sum_{i=j+1}^{255} P_t(i) \times c_1\right] \geq CT \\ T = \mathrm{Max}[j\,] \end{array}} \tag{2}$$

Step 4: if $T < 2^{c_0}$ or $T > 255 - 2^{c_1}$, the following procedure will be executed.

$$\boxed{\begin{array}{l} \textbf{Case } 1: T < \ 2^{c_0}, \\ \qquad\qquad T = 0, c_0 = \ c_1; \\ \textbf{Case } 2: T > 255 - 2^{c_1}, \\ \qquad\qquad T = 255 - 2^{c_1}; \end{array}} \tag{3}$$

After executing the above steps, the pre-processing procedure is accomplished.

2.2 The Embedding Procedure

The secret data stream S is embedded by roster scan order. Assume P_i is one pixel of F. The pixel value of P_i is P_{ix}. The order of P_i is expressed as $F = \{ P_i \mid i = 1, 2, 3, 4, \ldots., M \times N \}$. For each pixel, the detailed embedding steps are follows.

Input: a cover-image F', threshold pixel value T, c_0 and c_1.
Output: a stego-image F'.

Step 1: The k_i is the number of embedding bits of P_i which can be derived by following rules.

$$\boxed{\begin{array}{l} \textbf{Case } 1: P_{ix} \leq T, \\ \qquad\qquad k_i = \ c_0; \\ \textbf{Case } 2: P_{ix} > T, \\ \qquad\qquad k_i = \ c_1; \end{array}} \tag{4}$$

Step 2: Extract k_i secret bits to form S, and embed it into P_i by the modified LSB substitution method.

Step 3: Let P'_{ix} be the embedded result of P_{ix}. If P'_{ix} and P_{ix} belong to the same level, the embedding procedure of P_i is accomplished. Otherwise, the emendation actions should be executed as follows.

Repeat step 1–3 until the secret data stream S is embedded into F. The flowchart of Adaptive Threshold Generation Scheme and embedding procedures is shown in Fig. 1.

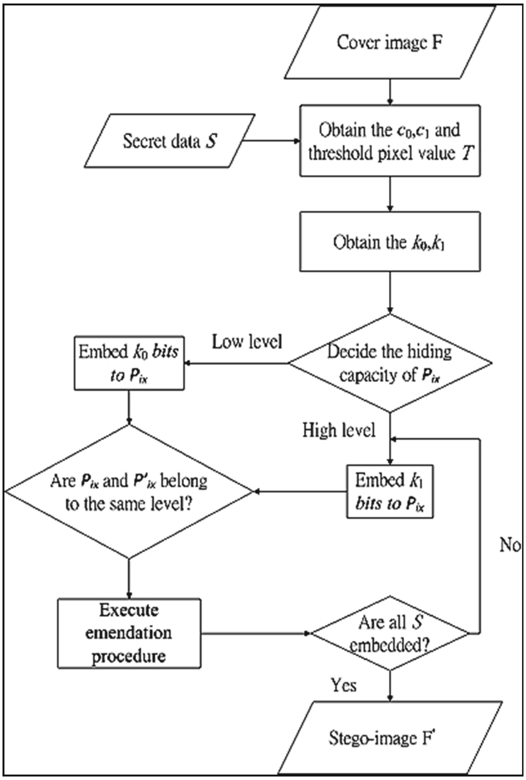

Cover image F
Secret data S
Obtain the c_0,c_1 and threshold pixel value T
Obtain the k_0,k_1
Low level
Decide the hiding capacity of P_{ix}
Embed k_0 bits to P_{ix}
High level
Are P_{ix} and P'_{ix} belong to the same level?
Embed k_1 bits to P_{ix}
No
Execute emendation procedure
Are all S embedded?
Yes
Stego-image F'

Fig. 1. Flowchart of the pre-processing and embedding procedures.

2.3 The Extracting Procedure

In the recovery process, the embedded data can be extracted rapidly from the stego image by the threshold pixel value T, the embedding capacity of higher and lower level k_0 and k_1. The scan order of extracting procedure is the same as embedding procedure. The steps of data extraction are as follows.

Input: a stego-image F', threshold pixel value T, k_0 and k_1.
Output: a bitstream secret data.

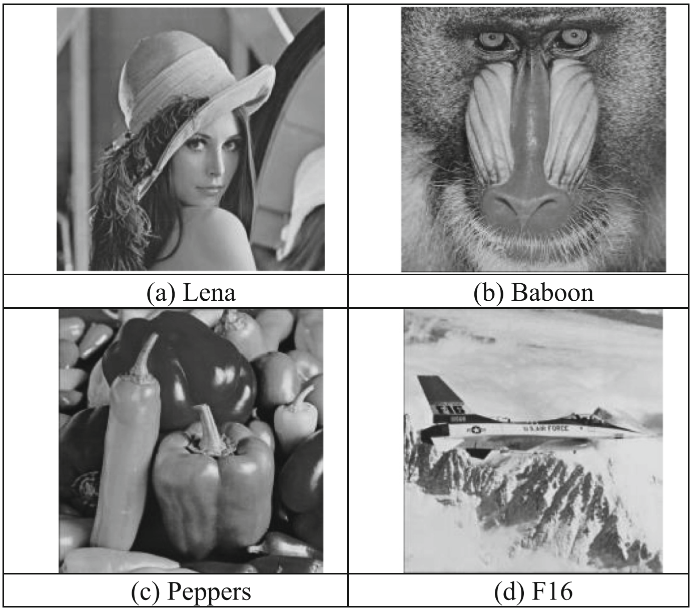
Step 1: If P'_{ix} is more than T, P'_{ix} belongs to higher level, and k_1-LSB secret data should be extracted. Otherwise, P'_{ix} belongs to lower level, and k_0-LSB secret data should be extracted.
Step 2: Repeat step 1 until all the secret data have extracted.
Step 3: Join all the pieces of secret data together and return the secret data stream S.

3 Experimental Results and Discussion

Experimental results of the proposed scheme are presented and discussed in this section. The experimental environment is using an Intel(R) Core(TM) 4950 computer with 8 GB of memory. The program is written in MATLAB program tool. The secret data stream S is generated by pseudo-random numbers. Four 512 × 512 gray-scale test images are used, namely “Lena”, “Baboon”, “Peppers” and “F16”, as shown in Fig. 2. The peak-signal-to-noise ratio (PSNR) is utilized to evaluate the distortion of the stego-image after the secret data have embedded. The PSNR is defined in following equation.

$$\mathrm{PSNR} = 10 \times \log \frac{255^2}{\mathrm{MSE}} (\mathrm{dB}), \tag{6}$$

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(E_{ij} - C_{ij} \right). \tag{7}$$



(a) Lena

(b) Baboon

(c) Peppers

(d) F16

Fig. 2. Four 512 * 512 gray-scale cover-images.

In Eq. (6), the value 255 is the maximum value of the pixel value. The mean square error (MSE) represents the distortion between the $M \times N$ cover image C and its stego-image E. The notations E_{ij} and C_{ij} represent the pixel values at the coordinate (*i, j*) of image *E* and *C*, respectively. In general, a large PSNR value indicates the dissimilarity is small between the cover image and the stego-image that is more imperceptible to the human eye.

3.1 Experimental Results

Tables 1, 2 and 3 summarize the comparison results with other spatial domain data hiding algorithm [4, 5, 9, 10] for four test images. Because the maximum hiding capacity of [9, 10] are less than 4 bpp, we adjust the hiding capacity of proposed method consistent with above schemes' maximum capacities, and the comparison results are displayed in Tables 1 and 2. Table 3 shows the comparison results with [4, 5], and all the hiding capacities are 4.3 bpp. From Tables 1, 2 and 3, although the methods listed in the tables are resulted in the same hiding capacity, the PSNR values of the proposed method outperform that of other methods, which is enhanced about 2 to 3 dB performance for the test images. Figures 3 and 4 show the visual quality comparison results between the proposed method and [4, 5, 9–11].

Table 1. The visual quality comparison results of the proposed method with [9].

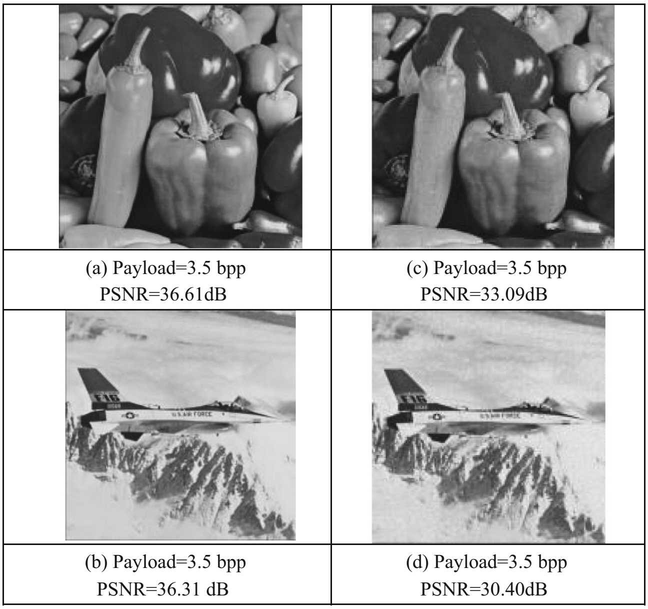
	The proposed method		PVD-LSB method [9]	
Cover image (T)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)
Lena (212)	528512	45.99	528512	38.94
Baboon (198)	544056	46.02	544056	33.43
Peppers (210)	528256	46.09	528256	37.07
F16 (196)	530048	47.73	530048	37.42

Table 2. The visual quality comparison results of the proposed method with [10].

	The proposed method		PVD-MOD method [10]	
Cover image (T)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)
Lena (120)	409752	47.80	409752	38.80
Baboon (122)	457168	47.76	457168	40.3
Peppers (85)	407256	47.83	407256	43.3
F16 (221)	409792	47.73	409792	43.5

Table 3. The visual quality comparison results of the proposed method with [4, 5] methods.

	The proposed method		[4] method	[5] method
Cover image (T)	Capacity (bits)	PSNR (dB)	PSNR (dB)	PSNR (dB)
Lena (120)	1127219 (4.3bpp)	30.84	26.48	29.47
Baboon (122)		31.02	26.46	29.46
Peppers (85)		30.89	26.34	29.18
F16 (221)		30.15	26.61	29.45



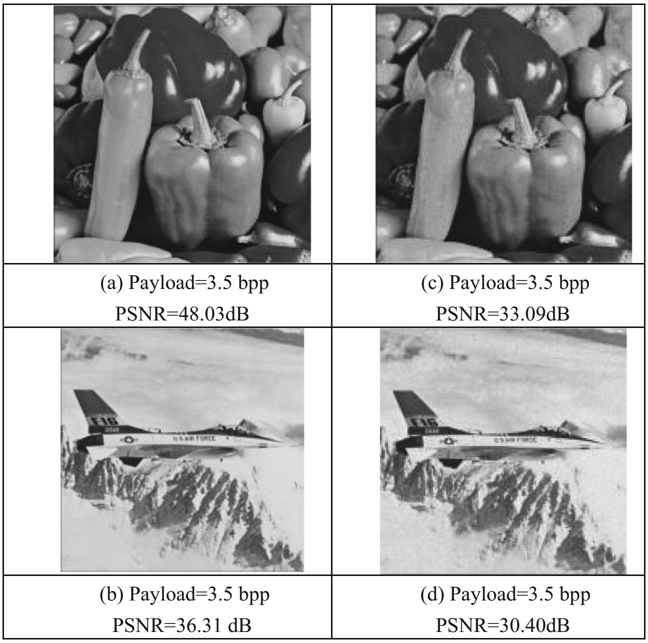
(a) Payload=3.5 bpp
PSNR=36.61dB

(c) Payload=3.5 bpp
PSNR=33.09dB

(b) Payload=3.5 bpp
PSNR=36.31 dB

(d) Payload=3.5 bpp
PSNR=30.40dB

Fig. 3. Comparison results of [9] and the proposed method; (a)–(b) the stego images generated by the proposed scheme and (c)–(d) the stego images generated by [9].



(a) Payload=3.5 bpp
PSNR=48.03dB

(c) Payload=3.5 bpp
PSNR=33.09dB

(b) Payload=3.5 bpp
PSNR=36.31 dB

(d) Payload=3.5 bpp
PSNR=30.40dB

Fig. 4. Comparison results of [10] and the proposed method s; (a)–(b) the stego images generated by the proposed scheme and (c)–(d) the stego images generated by [10].

3.2 Discussion

In order to obtain better stego-image, a data hiding algorithm has to minimize the volatility of each pixel value in the embedding procedure. The proposed method has a dynamic strategy to adjust hiding capacity consistent with the length of secret message, and the secret message can be embedded in stego-image as averagely as possible. For example, although the hiding capacity of all methods is 4.3 bpp in Table 3, the number of unchanged pixels of each method is different. The number of unchanged pixels generated by the proposed approach is 425, 327, 62, and 1022 in four stego-images, respectively, but other methods have a large number of unchanged pixels. As such, the secret message can be more averagely embedded on each pixel by the proposed method, and the quality of stego-image would be better.

4 Conclusion

In this paper, an adaptive pixel value threshold strategy scheme that boosting up the performance of LSB based steganography is proposed. The adaptive threshold strategy is based on hiding capacity requirement and cumulative statistics of cover-image's pixel value in different cases to select the appropriate threshold pixel value. The advantages of this paper are summarized as follows. First, this adaptive threshold strategy is proposed to devise an efficient adaptive steganographic method. Secondly, the proposed method improves the hiding capacity of [7, 9, 10] methods and the PSNR value is still above 30. Thirdly, it has higher PSNR value than [4–7, 9, 10] at 10 different hiding capacities. Fourthly, the stego-image generated by the proposed scheme has very few unchanged pixel, and it is difficult to be perceived by human eyes. Therefore, the proposed scheme has significant promotions in terms of capacity and imperceptivity, and it is applicable to practical steganographic application.

References

1. Highland, H.J.: Data encryption: a non-mathematical approach. Comput. Secur. **16**(5), 369–386 (1997)
2. Wang, H., Wang, S.: Cyber warfare: steganography vs. steganalysis. Commun. ACM **47**(10), 76–82 (2004)
3. Katzenbeisser, S., Petitcolas, F.A.: Information Hiding Techniques for Steganography and Digital Watermarking, hardcover edn. Artech House (2000)
4. Bender, W., Gruhl, D., Morimoto, N., Lu, A.: Techniques for data hiding. IBM Syst. J. **35**(3), 313–336 (1996)
5. Chan, C.-K., Cheng, L.-M.: Hiding data in images by simple LSB substitution. Pattern Recogn. **37**(3), 469–474 (2004)
6. Thien, C.-C., Lin, J.-C.: A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function. Pattern Recogn. **36**(12), 2875–2881 (2003)
7. Wu, D.-C., Tsai, W.-H.: A steganographic method for images by pixel-value differencing. Pattern Recogn. Lett. **24**(9–10), 1613–1626 (2003)
8. Yang, C.-H.: Inverted pattern approach to improve image quality of information hiding by LSB substitution. Pattern Recogn. **41**(8), 2674–2683 (2008)

9. Wu, H.-C., Wu, N.-I., Tsai, C.-S., Hwang, M.-S.: Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEE Proc. Vis. Image Sig. Process. **152**(5), 611–615 (2005)
10. Wang, C.-M., Wu, N.-I., Tsai, C.-S., Hwang, M.-S.: A high quality steganographic method with pixel-value differencing and modulus function. J. Syst. Softw. **81**(1), 150–158 (2008)
11. Yang, C.-H., Weng, C.-Y., Wang, S.-J., Sun, H.-M.: Adaptive data hiding in edge areas of images with spatial LSB domain systems. IEEE Trans. Inf. Forensics Secur. **3**(3), 488–497 (2008)
12. Lou, D.-C., Ho, S.-C., Chiu, C.-C.: Hybrid high-capacity data hiding by pixel-value differencing and modulus function. J. Internet Technol. **12**(3), 303–312 (2011)
13. Lee, Y.-P., Lee, J.-C., Chen, W.-K., Chang, K.-C., Su, I.-J., Chang, C.-P.: High-payload image hiding with quality recovery using tri-way pixel-value differencing. Inf. Sci. **191**, 214–225 (2012)
14. Yang, C.-H., Weng, C.-Y., Tso, H.-K., Wang, S.-J.: A data hiding scheme using the varieties of pixel-value differencing in multimedia images. J. Syst. Softw. **84**(4), 669–678 (2011)
15. Yang, C.-N., Hsu, S.-C., Kim, C.: Improving stego image quality in image interpolation based data hiding. Comput. Stand. Interfaces **50**, 209–215 (2017)
16. Hussain, M., Wahab, A.W.A., Ho, A.T.S., Javed, N., Jung, K.-H.: A data hiding scheme using parity-bit pixel value differencing and improved rightmost digit replacement. Sig. Process. Image Commun. **50**, 44–57 (2017)
17. Hong, W.: Adaptive image data hiding in edges using patched reference table and pair-wise embedding technique. Inf. Sci. **221**, 473–489 (2013)
18. Xu, W.-L., Chang, C.-C., Chen, T.-S., Wang, L.-M.: An improved least-significant-bit substitution method using the modulo three strategy. Displays **42**, 36–42 (2016)
19. Lu, T.-C.: Interpolation-based hiding scheme using the modulus function and re-encoding strategy. Sig. Process. **142**, 244–259 (2018)
20. Swain, G.: A steganographic method combining LSB substitution and PVD in a block. Procedia Comput. Sci. **85**, 39–44 (2016)

Shamir's Secret Sharing Scheme in Parallel

Shyong Jian Shyu(✉) and Ying Zhen Tsai

Department of Computer Science and Information Engineering,
Ming Chuan University, 5 Der Ming Road, Gui Shan, Taoyuan 333, Taiwan
sjshyu@mail.mcu.edu.tw, alonstilllove@gmail.com

Abstract. A (*k*, *n*) threshold secret sharing scheme encrypts a secret *s* into *n* parts (called shares), which are distributed into *n* participants, such that any *k* participants can recover *s* using their shares, any group of less than *k* ones cannot. When the size of *s* grows large (e.g. multimedia data), the efficiency of sharing/decoding *s* becomes a major problem. We designed efficient and parallel implementations on Shamir's threshold secret sharing scheme using sequential CPU and parallel GPU platforms, respectively, in a personal computer. Experimental results show that GPU could achieve an appealing speedup over CPU when dealing with the sharing of multimedia data.

Keywords: Secret sharing · Threshold scheme · Parallel computing

1 Introduction

Secret sharing provides the protection of a secret among a group of participants. A (*k*, *n*) threshold secret sharing scheme (TSSS) aims at encrypting a secret *s* into *n* parts (called shadows), which are distributed into *n* participants, such that any *k* participants can recover *s* using their shares, while any group of less than *k* ones cannot. In 1979, Shamir [1] proposed an elegant TSSS based on polynomial interpolation under Galois field. It is perfect and ideal. Since then, quite a lot researches devoted their works in this topic.

In Shamir's TSSS, secret *s* is merely a number. It could be successfully applied to share a key with a moderate data size. Nowadays, multimedia data, such as images, voices, video clips, etc., whose sizes are relatively larger than that of an ordinal number/key. The efficiency of the encoding and decoding processes in the TSSS becomes a major and significant consideration for practical usage.

Exploiting parallel computing to cope with the problem has been studied by Fang [2]. The parallel environment was constructed by multiple CPUs (Intel Xeon CPU 2.66 GHz) under Open MPI. In a 12 CPUs cluster, his parallel decoding performance receives about 7–10 times faster than that of using a single CPU for sharing a 512 × 512 image.

Recently, high performance computing resulting from GPGPU (general-purpose computing on graphics processing units, GPU for short) [3] has been a popular and

This research was supported in part by the Ministry of Science and Technology, Taiwan, under Grants MOST 103-2221-E-130-002-MY3.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 206–216, 2018.
https://doi.org/10.1007/978-3-319-76451-1_20

economic solution to time-consuming problems in many applications. Under CUDA (compute unified device architecture) environment [4], GPU consisting of up to thousands of cores in a single graphic card (or multiple cards) can be programmed to execute simultaneously in a massively parallel fashion using some high-level languages (e.g., C, Java, etc.).

In this paper, we develop efficient sequential and parallel algorithms, which are executed in sequential CPU and parallel GPU platforms, respectively, in a personal computer, to implement Shamir's TSSS concerning the multimedia data. Section 2 briefly reviews the Shamir's TSSS. Section 3 presents several sequential algorithms for this TSSS and Sect. 4 discusses our parallel algorithms which are suitable for running in GPU. The experimental results of our sequential and parallel implementations are summarized and compared in Sect. 5. Section 6 gives some concluding remarks.

2 Shamir's TSSS

Consider q, k, n and secret s for $k \le n < q$ and $0 \le s < q$ where $q = p^m$, in which p is a prime number and m is a positive integer. s is encoded into n shadows $y_1, y_2, \ldots, y_n$ for a set of n participants $P = \{1, 2, \ldots, n\}$ by the following polynomial function with $k-1$ degrees:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1} (\bmod\, q) \tag{1}$$

where a_0 is s and a_1, a_2, …, a_{k-1} are random numbers with $0 \le a_l < q$ for $1 \le l \le k - 1$. The dealer determines n distinct random keys x_1, x_2, …, x_n, constructs n shadows y_1, y_2, …, y_n and distributes (x_i, y_i) to participant i where $0 < x_i < q$ and $y_i = f(x_i)$ for $1 \le i \le n$. Note that all computations are with modular arithmetic under prime power q (i.e. GF(q)) with $q > n$ [1, 5].

In the decoding phase, any k participants, say i_1, i_2,…, i_k, with k pairs of keys and shadows $(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_k}, y_{i_k})$ can form k linear equations as follows.

$$\begin{aligned} y_{i_1} &= a_0 + a_1 x_{i_1} + a_2 x_{i_1}^2 + \ldots + a_{k-1} x_{i_1}^{k-1} \\ y_{i_2} &= a_0 + a_1 x_{i_2} + a_2 x_{i_2}^2 + \ldots + a_{k-1} x_{i_2}^{k-1} \\ &\cdots \\ y_{i_k} &= a_0 + a_1 x_{i_k} + a_2 x_{i_k}^2 + \ldots + a_{k-1} x_{i_k}^{k-1} \end{aligned} \tag{2}$$

Solving these equations, we can find all the k coefficients a_0, a_1, …, a_{k-1} of $f(x)$. s (= a_0) can thus be obtained. Specifically, to reveal secret s (i.e., coefficient a_0), we simply compute (3) as follows.

$$a_0 = \sum_{u=1}^{k} y_{i_u} \left(\prod_{v=1, v \ne u}^{k} \frac{-x_{i_v}}{x_{i_u} - x_{i_v}} \right) \tag{3}$$

Note that Shamir's TSSS is with perfect security. That is, any k-1 or less shadows gain no information about s. It is also ideal so that the size of each share is the same as that of the secret. Our realizations here preserve these two merits.

3 Sequential TSSS Algorithms

3.1 Sharing a Number

Shamir's TSSS may be realized in several ways. Since the input data is treated as the binary one, we choose $q = 2^m$, which makes the addition and subtraction operations in GF(q) much easier by adopting xor operation. For the encoding phase, a naive algorithm, denoted as E1, is presented in the following. Owing to the computations in GF (q), E1 adopts *irreducible polynomial* and *multiplicative inverse* to deal with the multiplication and division operations [5, 6].

```
Encoding
Input: $k$, $n$, secret $s \in [0, q)$ under prime power $q$ and a set of $n$ keys: $X = \{x_1, x_2, \ldots, x_n\}$
Output: $n$ shares: $Y = \{y_1, y_2, \ldots, y_n\}$
E1($k$, $n$, $s$, $q$, $X$)
1.      Generate $a_1, a_2, \ldots, a_{k-1}$ ($\in [0, q)$) randomly
2.      Set $a_0 = s$
3.      Let $f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{k-1}x^{k-1}$      // GF($q$)
4.      for (each $x_i$, $1 \le i \le n$)
        {       Compute $y_i = f(x_i)$ using irreducible polynomial and multiplicative
                inverse
        }
5.      return $Y = \{y_1, y_2, \ldots, y_n\}$
```

To reduce the complicated calculation time of multiplications and divisions in GF (2^P), we resort to two tables: Log and Antilog (with size q both). Specifically, the product of two values is the antilog of the mod (GF−1) sum of their logs, while the quotient of them is the antilog of the mod (GF−1) difference between their logs [5]. Exploiting Log and Antilog tables, multiplications and divisions are expected to be more efficient. Algorithm E2 conducts such idea.

```
E2($k$, $n$, $s$, $q$, $X$)
1-3.    Same as E1
4.      Prepare Log and Alog tables
5.      for (each $x_i$, $1 \le i \le n$)
        {     Compute $y_i = f(x_i)$ using xor for + and table-lookup for ×
        }
6.      return $Y = \{y_1, y_2, \ldots, y_n\}$
```

The computation of the polynomial in (1) can be accelerated by the famous Horner's scheme [7], which reduces the number of multiplications, as below.

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1} (\text{mod } q) \\ &= a_0 + x(a_1 + a_2 x + \ldots + a_{k-1} x^{k-2}) (\text{mod } q) \\ &= a_0 + x(a_1 + x(a_2 + \ldots + a_{k-1} x^{k-3})) (\text{mod } q) \\ &= \ldots \\ &= a_0 + x(a_1 + x(a_2 + \ldots + x(a_{k-2} + a_{k-1} x))) (\text{mod } q) \end{aligned} \tag{4}$$

Following the Horner's scheme and (4), we develop Algorithm E3 as follows.

```
E3($k$, $n$, $s$, $q$, $X$)
1-4.   Same as E2
5.     for (each $x_i$, $1 \le i \le n$)
       {    Compute $y_i = f(x_i)$ using xor for + and table-lookup for × with Horner's
            scheme
       }
6.     return $Y = \{y_1, y_2, \ldots, y_n\}$
```

For a further possible refinement, we may pre-compute x_i^a's for $1 \le a \le k-1$ and $1 \le i \le n$ when $x_1, x_2, \ldots, x_n$ is given. In algorithm E4, we prepare an $n \times (k-1)$ table, called Power table, to store the values of x_i^a's to possibly accelerate the computations for x_i^a's in $y_i = f(x_i)$ (Step 6 in E4).

```
E4($k$, $n$, $s$, $q$, $X$)
1-4.   Same as E2
5.     Prepare Power table
6.     for (each $x_i$, $1 \le i \le n$)
       {    Compute $y_i = f(x_i)$ using xor for + and table-lookup for × and power
            operations
       }
7.     return $Y = \{y_1, y_2, \ldots, y_n\}$
```

Regarding the decoding phase, we adopt algorithm D1, which utilizes Log, Alog and Power tables (since these are expected to deliver a better performance). Again, all computations are in GF(q).

```
Decoding
Input: $t$ pairs of keys, $q$ and set of shares: $T = \{(x_{i_1}, y_{i_1}), (x_{i_2}, y_{i_2}), \ldots, (x_{i_t}, y_{i_t})\}$
Output: secret $s$ (if $t \geq k$), or random file (otherwise)

D1($t$, $q$, $T$)
1.    $a = 0$
2.    for (each $u$, $1 \leq u \leq t$)
      {    $b = 1$
           for (each $v \neq u$, $1 \leq v \leq t$)
                $b = b \times \frac{-x_{i_v}}{x_{i_u} - x_{i_v}}$    // GF($q$)
           $a = a + y_{i_u} \times b$    // GF($q$)
      }
3.    return $a$
```

With a subtle rearrangement, (3) could be reformulated as

$$a_0 = (-1)^{k-1}(x_{i_1}x_{i_2}\ldots x_{i_t})\sum_{u=1}^{t}\frac{y_{i_u}}{x_{i_u}c_{i_u}} \pmod{q}$$

where

$$c_{i_u} = \prod_{v=1, v\neq u}^{t}(x_{i_u} - x_{i_v}) \text{ for } 1 \leq u \leq t. \tag{5}$$

It is seen that the number of multiplications in (3) is reduced in (5). Algorithm D2 exploits this advantage.

```
D2($t$, $q$, $T$)
1.    $a = 1$
2.    for (each $u$, $1 \leq u \leq t$)
      {    $c_{iu} = 1$
           for (each $v \neq u$, $1 \leq v \leq t$)
                $c_{i_u} = b \times (x_{i_u} - x_{i_v})$    // GF($q$)
           $a = a + y_{i_u} \times (x_{i_u} \times c_{i_u})^{-1}$    // GF($q$)
      }
3.    $a = a \times (-1)^{k-1}(x_{i_1}x_{i_2} \ldots x_{i_t})$    // GF($q$)
4.    return $a$
```

3.2 Sharing a Binary Data

In order to share a binary data D with a size of N bytes, we decompose it into $\lambda = \lceil N/l \rceil$ segments with l bytes each (or (N mod l) bytes for the last segment if N is not divisible by l). The general algorithm $S(alg, k, n, D, q, X, T, N, l)$ for encoding/decoding a binary data is an abstraction for all algorithms introduced in 3.1 where $alg \in$ {E1, E2, E3, E4, D1, D2} and the parameters may be null (or an empty set) if alg does not require such parameters.

```
Encoding and Decoding
Input: $k$, $n$, secret $D$ with $N$ bytes, $q$, $alg \in$ {E1, E2, E3, E4, D1, D2}, $X = \{x_1, x_2, \dots, x_n\}$ for $alg \in$ {E1, E2, E3, E4} and $\mathcal{T} = \{(x_{i_1}, Y_{i_1}), (x_{i_2}, Y_{i_2}), \dots, (x_{i_t}, Y_{i_t})\}$ ($t = |\mathcal{T}|$) for $alg \in$ {D1, D2}
Output: $\mathcal{Y} = \{Y_1, Y_2, \dots, Y_n\}$ for $alg \in$ {E1, E2, E3, E4} or $\mathcal{D} = \{\mathscr{d}_1, \mathscr{d}_2, \dots, \mathscr{d}_\lambda\}$ for $alg \in$ {D1, D2}
-----------------------------------------------------------------
$S(alg, k, n, D, q, X, \mathcal{T}, N, l)$
1.  $\lambda = \lceil N/l \rceil$
2.  if ($alg \in$ {E1, E2, E3, E4})
    {   Decompose $D$ into $\lambda$ segments: $d_1, d_2, \dots, d_\lambda$
        for (segment $d_i$, $1 \le i \le \lambda$) $(y_{1i}, y_{2i}, \dots, y_{ni}) = alg(k, n, d_i, q, X)$
        for (each participant $i$, $1 \le i \le n$) $Y_i = y_{i1} \cup y_{i2} \cup \dots \cup y_{i\lambda}$
        return $\mathcal{Y} = \{Y_1, Y_2, \dots, Y_n\}$
    }
3.  for ($1 \le i \le \lambda$) $\mathscr{d}_i = alg(t, q, \mathcal{T})$        // $\mathscr{d}_i = d_i$, if $t \ge k$
    return $\mathcal{D} = \mathscr{d}_1 \cup \mathscr{d}_2 \cup \dots \cup \mathscr{d}_\lambda$        // ($\mathcal{D} = D$ if $t \ge k$)
```

4 Parallel Algorithms for Shamir's TSSS

To utilize the capability of parallel execution among the cores in GPU, we decompose the data into fine grains such that each grain could be handled by a thread in GPU. Consider binary data D with a size of N bytes, a word size of l bytes, and GPU with a maximal number $maxT$ of conceptually simultaneous running threads. Let $\lambda = \lceil N/l \rceil$. Since λ may be larger than $maxT$, we further partition these λ segments into $\eta = \lceil \lambda / maxT \rceil$ regions $e_1, e_2, \dots, e_\eta$ with $maxT$ segments each (or (λ mod $maxT$) segments for the last region if λ is not divisible by $maxT$). The $maxT$ segments are distributed to the cores in GPU, which are running in parallel. We call it a *run*. The whole task would be finished after all $\lceil \lambda / maxT \rceil$ runs.

In the encoding phase, each segment requires $k-1$ random numbers for its corresponding polynomial. In a single run, we need $(k-1)maxT$ random numbers and, in total, $\lambda(k-1)$ ones are required. This job can also be realized in parallel in GPU.

```
Parallel Encoding
Input: $k$, $n$, secret $D$ under prime power $q$ and a set of $n$ keys: $X = \{x_1, x_2, \ldots, x_n\}$
Output: $n$ shares: $Y = \{Y_1, Y_2, \ldots, Y_n\}$
$PE(k, n, s, q, X, N, l, maxT)$
    $\lambda = \lceil N/l \rceil$
    $\eta = \lceil \lambda/maxT \rceil$
    Allocate memory of $Y_j$ for $1 \le j \le n$ in CPU and GPU
    for (each region $e_i$, $1 \le i \le \eta$)
    {   __paralleldo<<$maxT$>>__
            $\{y_{1i}, y_{2i}, \ldots, y_{ni}\} = Encode(k, n, e_i, q, X)$
            copy $\{y_{1i}, y_{2i}, \ldots, y_{ni}\}$ back to CPU
        __parallelend<<$maxT$>>__
    }
    for (each participant $i$, $1 \le i \le n$) $Y_i = y_{i1} \cup y_{i2} \cup \ldots \cup y_{i\eta}$
    return $Y = \{Y_1, Y_2, \ldots, Y_n\}$
```

Note that the area between __paralleldo <<*maxT*>>__ and __parallelend <<*maxT*>>__ is running in parallel among *maxT* threads in GPU. Note that the actual number of the cores in GPU depends on the hardware specification, whereas, *maxT* is the number of threads conceptually. In our test platforms, *maxT* is 65535 × 1024. Definitely, we shall choose the most significant algorithm among E1, E2, E3 and E4 to be the *Encode* function in Algorithm *PE*. Similarly, the most efficient one between D1 and D2 is chosen as the *Decode* function in Algorithm *PD* as follows.

```
Parallel Decoding
Input: $t$ pairs of keys and shares: $\mathcal{T} = \{(x_{i_1}, Y_{i_1}), (x_{i_2}, Y_{i_2}), \ldots, (x_{i_t}, Y_{i_t})\}$
Output: secret $D$ (if $t \ge k$), or random file (otherwise)
$PD(t, q, \mathcal{T}, N, l, maxT)$
    $\lambda = \lceil N/l \rceil$
    $\eta = \lceil \lambda/maxT \rceil$
    Allocate memory of $\mathcal{D}$ in CPU and GPU
    for (each region $e_i$, $1 \le i \le \eta$)
    {   __paralleldo<<$maxT$>>__
            $\{d_1, d_2, \ldots, d_t\} = Decode(maxT, t, q, \mathcal{T})$
            copy $\{d_1, d_2, \ldots, d_t\}$ back to CPU
        __parallelend<<$maxT$>>__
    }
    return $\mathcal{D} = d_1 \cup d_2 \cup \ldots, d_t$        // $\mathcal{D} = D$, if $t \ge k$
```

5 Experimental Results

We tested the aforementioned sequential and parallel algorithms in CPU and GPU platforms respectively. The CPU platform consists of an i7-4790 (3.6 GHz) processor and 8 GB RAM under a PC running Windows 7. The programs were coded in Borland C++ Builder. The same PC with a GeForce GTX 760 video card which owns 1152 cores and 2 GB RAM acts as the GPU platform 1. Another GTX Titan X card consisting of 3072 cores and 12 GB RAM acts as the GPU platform 2. The parallel CUDA programs were coded in Visual Studio 2015.

First of all, we would like to know the performances of Algorithm *S* using E1 and E2, respectively, as parameter *alg*. A binary file with 7.8 MB in a (4, 5) threshold structure was tested. Table 1 reveals the encoding times in seconds using $l = 1$ and $m = 8$. It is seen from Table 1 that E2 is much faster than E1 in this case. The advantage of E2 over E1 (which applies complicated arithmetics for addition and multiplication) is quite intuitive.

Table 1. Comparison of Algorithm *S* using E1 and E2

	Encoding
E1	89.74
E2	4.23
E1/E2	21.22

Then, we compare the performances of Algorithms E2, E3 and E4 for a 15.9 MB binary file with (k, n) = (2, 5), (4, 11), (6, 21) and (8, 31). The execution times in seconds are presented in Table 2. It is seen that both E3 and E4 are superior to E2. Further, E3 is a bit better than E4 for (2, 5), while E4 is slightly better than E3 for the other cases. The Power table improves the calculation of polynomials required in the encoding process. Comparing the values of E2/E4 ratios, we realize that when k and n grow, the superiority of E4 over E2 tends to grow. Since the Horner's scheme in E3 only delivers a better performance for small cases like (2, 5), we adopt E4 as the sequential encoding algorithm for the following experiments.

Table 2. Comparison of Algorithm *S* using E2, E3 and E4

(k, n)	(2, 5)	(4, 11)	(6, 21)	(8, 31)
E2	2.31	18.00	71.89	185.00
E3	0.99	5.43	16.94	36.01
E4	1.15	5.39	15.38	35.13
E2/E4	2.01	3.34	4.67	5.27
E3/E4	0.86	1.01	1.10	1.03

Concerning the decoding phase, the comparison between the decoding times of D1 and D2 in seconds is given in Table 3. It is seen that D2 is better than D1. These outcomes demonstrate the effectiveness of (4) for improving the computation in decoding. Note that we set parameter t as k in these test cases.

Table 3. Comparison of Algorithm S using D1 and D2

(k, n)	(2, 5)	(4, 11)	(6, 21)	(8,31)
D1	1.06	4.40	11.25	20.64
D2	0.70	1.90	3.99	6.97
D1/D2	1.51	2.32	2.82	2.96

Based on the above computational results, we obtain that Algorithm S using E4 and D2, denoted as S(E4) and S(D2), are more efficient than the other alternatives for encoding and decoding, respectively. In the following, we compare the performances of these two algorithms against those of *PE* using E4 and *PD* using D2, respectively, under GTX 760 and Titan X, denoted as $PE4_{760}/PE4_{\mathrm{X}}$ and $PD2_{760}/PD2_{\mathrm{X}}$.

Table 4 exhibits the computational results in seconds of S(E4) against $PE4_{760}$ and $PE4_{\mathrm{X}}$, and S(D2) against $PD2_{760}$ and $PD2_{\mathrm{X}}$ for various (k, n)'s with the same 15.9 MB data. When k is fixed, it is easily seen from Table 4 that the encoding time grows as n increases for both CPU and GPU. This is reasonable since a larger n induces more

Table 4. Results of S(E4) against $PE4_{760}$ and $PE4_{\mathrm{X}}$, and S(D2) against $PD2_{760}$ and $PD2_{\mathrm{X}}$

(k, n)	Encoding					Decoding				
	S(E4)	$PE4_{760}$	$PE4_{\mathrm{X}}$	S(E4)/ $PE4_{760}$	S(E4)/ $PE4_{\mathrm{X}}$	S(D2)	$PD2_{760}$	$PD2_{\mathrm{X}}$	S(D2)/ $PD2_{760}$	S(D2)/ $PD2_{\mathrm{X}}$
(2, 5)	1.15	0.21	0.15	5.57	7.95	1.42	0.06	0.04	22.17	37.69
(2, 11)	2.31	0.31	0.21	7.47	11.16	1.44	0.07	0.03	20.21	42.67
(2, 21)	4.31	0.42	0.32	10.35	13.42	1.44	0.07	0.03	22.08	42.87
(2, 31)	7.15	0.60	0.44	12.01	16.12	1.44	0.07	0.04	22.08	38.18
(2, 43)	10.17	0.81	0.60	12.62	16.99	1.45	0.07	0.04	20.73	38.03
(4, 5)	2.68	0.37	0.18	7.31	14.99	3.90	0.17	0.11	22.54	36.74
(4, 11)	5.40	0.57	0.32	9.45	17.07	3.87	0.17	0.10	22.63	39.48
(4, 21)	10.06	0.74	0.53	13.56	18.94	3.85	0.17	0.11	22.27	34.29
(4, 31)	15.24	0.98	0.75	15.55	20.35	3.88	0.17	0.10	22.71	39.56
(4, 43)	21.20	1.26	0.98	16.79	21.54	3.90	0.17	0.13	22.81	31.11
(6, 11)	8.28	0.81	0.46	10.23	17.89	8.00	0.31	0.24	25.82	33.86
(6, 21)	15.38	1.16	0.79	13.31	19.45	8.10	0.33	0.24	24.68	33.89
(6, 31)	23.09	1.51	1.09	15.32	21.23	8.32	0.32	0.25	26.07	33.04
(6, 43)	31.90	1.93	1.47	16.50	21.63	8.19	0.32	0.26	25.51	30.96
(8, 11)	12.45	1.06	0.59	11.73	21.06	13.90	0.52	0.35	26.78	39.96
(8, 21)	23.24	1.51	1.01	15.44	23.03	13.99	0.51	0.38	27.44	36.59
(8, 31)	35.13	1.98	1.43	17.71	24.54	14.29	0.51	0.43	28.13	33.14
(8, 43)	48.27	2.57	1.94	18.81	24.82	14.37	0.52	0.43	27.79	33.15

encoding computations for a fixed k. When n is fixed, such a tendency remains as k increases. $PE4_{760}$ ($PE4_{\mathrm{X}}$) delivers a speedup of about 5–18 (7–24) over S(E4) for different (k, n)'s in these experiments. Particularly, for (k, n) = (8, 21), S(E4) is 15.44 (23.03) times slower than $PE4_{760}$ ($PE4_{\mathrm{X}}$). Further, $PE4_{\mathrm{X}}$ is about 1.5 times faster than $PE4_{760}$. Regarding the decoding phase, the times for various n under a fixed k are almost the same since we set parameter t as k. The time increases as k grows under a fixed n. $PD2_{760}$ ($PD2_{\mathrm{X}}$) delivers a speedup of about 22–28 (30–42) over S(D2) for different (k, n)'s in our experiments. For (k, n) = (8, 21), S(D2) is 27.44 (36.59) times slower than $PD2_{760}$ ($PD2_{\mathrm{X}}$). Further, $PD2_{\mathrm{X}}$ is about 1.3 times faster than $PD2_{760}$.

To realize whether our parallel algorithm is applicable in real applications, we tested a 231.7 MB video clip on Titan X for (k, n) = (3, 8). It takes 3.31 (0.83) s in encoding (decoding for $t = k = 3$). These outcomes are quite acceptable for sharing large data in cloud storage service or private distributed database.

6 Concluding Remarks

Shamir's TSSS is graceful and elegant in the theoretical point of view. However, in practical implementation, it demands a thoughtful design. We present several realizations in sequential and in parallel. Apparently, skills such as preprocessing, table lookup and Horner's scheme provide a satisfactory performance to Shamir's TSSS in both the sequential and parallel algorithms.

The performances of our algorithms in the parallel platforms are more efficient than those of the sequential ones. The parallel computational results are appealing and significant since they demonstrate that applying secret sharing in the real applications is positively feasible. The i7 CPU for our sequential platform and Titan X for parallel are contemporary. Still, GTX 760, which is not an up-to-date device, produces a moderate performance with a lower cost (as compared to Titan X). As compared to the platforms of PC clusters or supercomputers, both Titan X and GTX 760 are more cost-effective.

More experiments would be conducted to obtain more clues to clarify the relationship among different k, n, l, m and larger data sizes. The concept of ramp secret sharing [8], which provides a trade-off between the level of security and the coding efficiency (or the share size), is worthy of further implementations if the share size becomes a compulsory consideration for practical applications.

References

1. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
2. Fang, W.-P.: Parallel processing for secret image sharing. In: International Symposium on Parallel and Distributed, IEEE Proceeding on Processing with Applications (ISPA10), Taipei, Taiwan, pp. 392–396 (2010)
3. Nvidia, Cuda GPUs. https://developer.nvidia.com/cuda-gpus
4. Nvidia, Cuda Zone. https://developer.nvidia.com/cuda-zone
5. Stallings, W.: Cryptography and Network Security Principles and Practices, 4th edn. Prentice Hall, Upper Saddle River (2005)

6. Silverman, J.H.: Fast multiplication in finite fields GF(2^N). In: Koç, Ç.K., Paar, C. (eds.) CHES 1999. LNCS, vol. 1717, pp. 122–134. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48059-5_12
7. Pankiewicz, W.: Algorithm 337: calculation of a polynomial and its derivative values by Horner scheme. Commun. ACM **11**(9), 633 (1968)
8. Blakley, G.R., Meadows, C.: Security of ramp schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 242–268. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_20

Network Security and Applications

A Cognitive Global Clock Synchronization Protocol in WSNs

Bilal Ahmad[1,2], Ma Shiwei[1](✉), and Fu Qi[1]

[1] Shanghai Key Laboratory of Power Station Automation Technology, School of Mechatronics Engineering and Automation, Shanghai University, No. 149, Yanchang Road,
Shanghai 200072, China
{abilali,masw}@shu.edu.cn, 18818218275@163.com

[2] Department of Electrical Engineering, The University of Poonch Rawalakot,
Rawalakot, AJ&K, Pakistan

Abstract. Clock synchronization is a crucial issue for data fusion, localization, duty cycle scheduling, and topology management in wireless sensor networks (WSNs). In this paper we proposed a cognitive global clock synchronization protocol (CGCSP) that is an accurate, energy efficient and reliable clock synchronization protocol in WSNs based on a single reference node. The CGCSP tackles the disadvantages of being single reference node by a cognitive switchover mechanism. This structure has been validated through the development of basic synchronization schemes i.e. sender-receiver (S-R) synchronization and receiver-receiver (R-R) synchronization. By evaluating and comparing the performances of it with state of the art protocols such as reference broadcast synchronization (RBS) and timing-sync protocol for sensor networks (TPSN), the proposed CGCSP shows reasonable lead over them in single hop network as well as in multi hop networks in terms of average synchronization accuracy and energy efficiency.

Keywords: Clock synchronization · Average synchronization accuracy
Energy efficiency

1 Introduction

WSNs are widely used in almost all type of applications such as military [1], environmental [2], industry [3], commercial and transportation [4]. Accomplishing all the benefits from WSNs clock synchronization is the backbone for all applications.

Clock synchronization is the basic feature for wired networks and wireless sensor networks (WSNs) for data synthesis and meaningful execution of any application. Pioneer work in the wired network was [5] and accurate time synchronization is under discussion in automation systems [6, 7]. However time synchronization in wireless sensor nets is quite different from those of the wired networks due to the resources constraints (limited energy at the end sensor node, less bandwidth, small processor, low quality quartz crystal etc.) in WSNs. Clock synchronization is the fundamental problem of the WSNs due to the above mentioned limitations. Local clocks of the sensor nodes must be synchronized not only because of specific application requirements but for the

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 219–231, 2018.
https://doi.org/10.1007/978-3-319-76451-1_21

channel access also [8]. Global positioning system is a high end practical solution. It requires expensive hardware circuitry to obtain accurate time synchronization with satellites. So it is preferred to synchronize the local clocks of the sensor nodes by messages exchange mechanism between them. Nodes use crystals to count the frequencies and various factors (aging, temperature, environmental factors etc.) affect it as a result local clocks of the nodes run at different rate so there is natural difference in the clock of two nodes [9].

Target tracking has gained considerable attention for its application in different fields such as military, civilian, and wildlife monitoring but getting accurate time synchronization in the presence of non-deterministic delays is very difficult. The delays are categorized as send time, access time, transmission time, propagation time and reception time [10, 11] and these uncertainties are minimized by MAC layer time stamping [10, 12]. As communication is the basic source of energy consumption [19] so the protocols with low communication cost are more energy efficient.

2 Problem Formulation

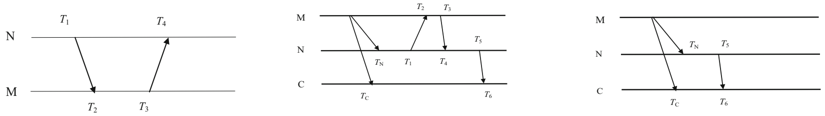
In the past one decade many synchronization schemes have been introduced to agree local clocks of the sensor nodes on the same time. Few of them adopt global clock synchronization like TPSN [10], RBS [11], and FTSP [12] and other are distributed schemes like [13–16]. In global clock synchronization single reference node is responsible for the time synchronization in the network while in distributed clock synchronization nodes reach on global time consensus through local exchange of information. Both schemes have advantages and disadvantages. The prior scheme is easy to control but in case of the reference node's failure clock synchronization is disturbed while in the latter case as there is no parent node so failure is not an option but topological control of the second scheme is really hard.

If the problem of the master node's failure is sorted out then maximum benefits can be achieved from the centralized synchronization schemes. The CGCSP exploits centralized synchronization method to achieve high accuracy because it uses reference clock of a single sensor node. But failure of the master node can cause malfunctioning in clock synchronization so a cognitive factor of expected master node is introduced in the global clock synchronization structure. This expected master node shares the value of computed clock offset to the child nodes and in this way physical clocks are converted into logical clocks. In CGCSP any node in the sensor network can be the master node provided that the node preserves highest energy level in the network and the node with second highest energy level is assigned the position of expected master node. CGCSP adopts centralized synchronization scheme to equalize $Ci(t)$ for all $i = 1, \ldots \ldots, N$. As with the passage of time local clocks of the sensor nodes drift from each other so CGCSP repeatedly corrects the phase offset to keep the clocks synchronized.

WSNs exhibit multi hop operations. Some existing protocols do not achieve good synchronization accuracy even for single hop network. Time synchronization protocols must achieve good synchronization accuracy even in multi hop networks while consuming least amount of energy. Many applications require synchronization accuracy less than 1 us for the significant achievement of their operations. RBS provides average accuracy of 29.1 us while TPSN shows 16.9 us accuracy for the single hop network and the case is worse for the multi hop network, FTSP produces the average accuracy of 0.5 us [20] while RTSP produces 0.25 us.

To keep time uncertainty in limits synchronization period must be performed periodically to renew the latest timing information of the node(s). Switchover mechanism is applied in CGCSP to overcome the problem of the failure of the master node (either malfunctioning or power drain) then a new pair of nodes starts acting as super nodes. Although logical clocks will not be synchronized during switchover period but as soon as switchover period ends clocks are resynchronized.

Although root/reference based protocols are topology sensitive but due to centralized mechanism it is easy to handle them and if root node election is done on regular basis then failure of master node is not a threat anymore. Existing solutions for time synchronization in WSNs are not resilient to the malicious nodes, CGCSP provide solution for it. Network wide synchronization is achieved in CGCSP with a master node as the root node. Every node synchronizes its local clock with the master node, so all nodes get synchronized with the master node. Messages exchange mechanism for the CGCSP is shown in Fig. 1.



(a) Synchronization between M and N. (b) Synchronization between N and C. (c) Integrated synchronization between M, N and C.

Fig. 1. Clock synchronization mechanism of the CGCSP.

The major innovation of CGCSP over other protocols is the dynamic selection of master node on the basis of energy level. Because if the compromised node becomes root node it shares false information and the child nodes calculate false offset and skew. So selection of master node on the energy basis saves from this adversary attacks.

A corrupted master node could send erroneous reference broadcast messages that would cause nodes to calculate wrong skew and offset. To avoid this we propose that instead of allowing a single node to be the master node for global time synchronization, any node can act as master node provided that it has the highest energy level. In the absence of any malicious nodes CGCSP is highly accurate and energy efficient and in the other case it is still accurate but little higher communication overhead affects energy efficiency.

3 Main Strategies of CGCSP

In WSNs applications the master node is likely to die due to the heavy workload of data transmission because the node runs out of energy. To overcome this problem, CGCSP method proposes that instead of allowing a single node to be the master node for global clock synchronization, any node can act as master node provided that it has the highest energy level.

In CGCSP, a cognitive switch-over mechanism is applied to ensure that when the master node is failure (either malfunctioning or power drain) a new node should start acting as the master node. By using this mechanism, although the logical clocks will not be synchronized during the switch-over period, the global clock can be re-synchronized as soon as the switchover period ends.

3.1 Synchronization Process of CGCSP

In order to keep the accuracy of time synchronization, sensor nodes must operate clock synchronization periodically to update the latest timing information.

The CGCSP exploits centralized synchronization method to achieve high accuracy by using reference clock of a single sensor node. Since local clocks of the sensor nodes drift from each other with the passage of time, the CGCSP repeatedly corrects the phase offset to keep the clocks synchronized. As depicted in Fig. 1, the clock synchronization process of CGCSP can be separated into two main phases. The first one is synchronization between '*M*' and '*N*', and the second one is synchronization between '*N*' and '*C*'.

In the first phase, the clock synchronization between node *M* and *N* use sender-receiver (S-R) synchronization scheme. Where T_1 refers to the local time of node *M* when it broadcasts to node *N*, T_2 refers to the local time of node *N* when it receive the message from node *M*, T_3 means the local time of node *N* when it transmits ACK (acknowledgement) message to node *M*, and T_4 means the local time of node *M* when it receives the ACK. In this phase, as the start of synchronization, node broadcasts synchronization message to all nodes in network. The node *N* receives and exchanges message with the node *M* and calculates its clock offset with the node *M*, and shares this message to the other child nodes. The clock offset between node *N* and *M* can be computed by:

$$offset(N, M) = \frac{(T_2 - T_1) - (T_4 - T_3)}{2} \tag{1}$$

Then, the node *N* compensates and updates its local clock by:

$$time\_value_N = time\_value_N - offset(N, M) \tag{2}$$

In the second phase, the clock synchronization between node *N* and node *C* uses receiver-receiver (R-R) synchronization model. In this process, when node *M* broadcasts synchronization message to nodes, node *C* also receives this information and compares it with the message received from node *N*. Then, the node *C* calculates its clock offset with node *M* and compensates and updates its local clock by following the formula:

$$offset(C, M) = offset(C, N) + offset(N, M) \tag{3}$$

$$offset(C, N) = T_C - T_N \tag{4}$$

Where T_C refers to the local time of node C when it hears the broadcast message from node M, and T_N refers to the local time of node N when it receives the broadcast message from node M.

Combining the above two phases, the integrated clock synchronization process of thee three nodes for CGCSP can be readily obtained, which is presented in Fig. 1(c).

Considering the energy consumption during the clock synchronization process of CGCSP, the more synchronization message is exchanged, the longer the synchronization time is spent, and the greater the energy is consumed by sensor nodes and the lower the efficiency of clock synchronization. In general, equation below can be used to estimate a radio signal's power which fades as it travels further from a transmitter [17]:

$$p_r = \frac{p_t}{d^c} \tag{5}$$

Where p_t is the transmitted power, d is the distance of the signal traveled from the transmitter, p_r refers to the received power of this signal after it traveled this distance, and c is the path loss coefficient. In practical applications of WSNs, the signal fades due to the diffractions, reflections, and refractions of walls and foliage so the path loss coefficient is often large.

Therefore, the energy consumption of a node during the clock synchronization process can be estimated in terms of messages transmission. According to the minimum transmission energy consumption model in the typical planar topology control algorithm, the energy consumption of the wireless sensor nodes is mainly related to the message bytes and the path [18]. So, we use the following formula to estimate the energy consumption of each synchronization message sent by the wireless sensor node.

$$E_t = 2.E_{elec}.k + E_{amp}.k.d^c \tag{6}$$

Where E_t is the energy consumption of a node to transmit a synchronization message, k is byte length of message, E_{elec} indicates the energy consumption of electronical devices and E_{amp} of transmitter amplifier of the node, d refers to the distance between the transmitting node and the receiving node. For the path loss coefficient c the general values are taken within 2–5. In the actual situation, it is generally taken 2 and for the outdoor environment the appropriate value is 4.

4 CGCSP Implementation

The CGCSP adopts following set of rules.

1. Create n wireless sensor nodes which form a WSN. In accordance with the CGCSP concept, each node contains the following information: node's ID (which is the node's coordinates during simulation), initial clock value and initial energy value.
2. All nodes are logically arranged in descending order of their initial energy values. The node with the highest initial energy value is chosen as the master/reference node

M, and the node with the second highest initial energy value is chosen as the expected master/reference node N.

3. Node M broadcasts synchronization message (its local clock value) to other nodes at time T_M. Node N receives this message at time T_N, and the other child nodes C receive this message at time T_C. Then, they prepare to start the global clock synchronization.
4. Node N first performs clock synchronization with node M by sending a synchronization request message to node M at time T_1. After receiving this request at T_2, node M feedbacks an ACK message (including T_2 and T_3 information) to node N at time T_3. Then, node N receives this ACK signal at T_4. At the same time, node N updates its local clock by calculating the $offset(N, M)$ using Eqs. (1) and (2).
5. Node N shares $offset(N, M)$ and T_N with other node C. Node C compares T_C with T_N and updates its local clock by calculating $offset(C, M)$ and $offset(C, N)$ using Eqs. (3) and (4).
6. Each time the message is transmitted; the node(s) will estimate its energy consumption according to the Eq. (6) and update its own energy value.
7. After the completion of clock synchronization, each node sends a message packet to node M, which includes node's ID, updated clock value and the energy value. Node M builds and updates a sensor nodes' information table for the whole network by using these message packets.
8. After receiving the clock information of all nodes in the network, node M judges whether its energy is lower than that of node N. If so, exchange the roles of master node M and expected master node N happens, preparing for the next clock synchronization period. At the same time if the new node N has the second highest energy value in all nodes is considered as expected master node. If not, re-select the expected master node.
9. Enter the next clock synchronization period and repeat this procedure from step 3.

For a multi-hop network, CGCSP will join the selection of intermediate nodes in each broadcast region. In the first broadcast region, the behaviors are carried out as the above 9 steps. In the second broadcast region, first chose the node i.e. most far away from the master node in first hop as an intermediate node, and then use it as the expected master node to start clock synchronization of this region from the above step (5). By analogy, in the third broadcast region, first select the node i.e. most far away from the intermediate node in second hop as a new intermediate node, and then use it as the new expected master node to start clock synchronization of this region from the above step (5) and so on to complete the clock synchronization of next broadcast region.

5 Simulations and Performance Evaluation

5.1 Simulation Setup

Simulations are carried out through MATLAB. During the simulation, the computational models are established for logical simulation to each algorithm regardless of real physical channel conditions and other contents of real WSNs. The CGCSP is implemented as a program to operate clock synchronization procedure in a mode of WSN. In order to simulate the message transmission process in clock synchronization, random error is joined to simulate the situations of time delay, packet loss and so on. In order to analyze the influence of the network size and multi-hop network on the performance of the CGCSP, simulation experiments are carried out for the networks with different node scales and ranging from single-hop to 5-hops.

In the simulation environment, a stochastic model of wireless sensor network (WSN) is established by randomly distributed 50–100 sensor nodes, such as shown in Fig. 2 for an instance of 100-nodes. In the simulated WSN model, a node's ID is represented by the coordinates of the node and the communication distance of two nodes is calculated by the coordinate distance between these two points. In order to observe the effect of the change of the number of messages exchange on synchronization performance, the network density in each experiment is controlled to ensure the effect of only one variable. In the experiments the network density (the ratio of network synchronization area and the number of nodes) is set to 1.5.

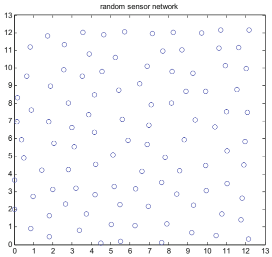


Fig. 2. Simulation model of a WSN composed of randomly distributed 100 nodes.

By taking the standard physical time *t* as the reference clock of each node, the initial clock value and initial clock skew are randomly assigned to each node. Assume that the message transmission time *delaytime* is a random variable related to the path length and the time when a node is ready to send ACK message is a random value within 1, as:

$$delaytime = random(1) \times distance \tag{7}$$

$$readytime = random(1) \tag{8}$$

The performance evaluation covers three aspects; synchronization accuracy, energy efficiency and comparison of CGCSP with other typical protocols used for WSNs. So the average synchronization error of 50–500 sensor nodes is collected for single hop and multi hop network. The energy consumption is observed during the synchronization period in whole network in terms of messages exchange. The performance of CGCSP is compared with that of RBS and TPSN in terms of synchronization errors and energy consumption. The simulation parameters are listed in the Table 1.

Table 1. Simulation parameters.

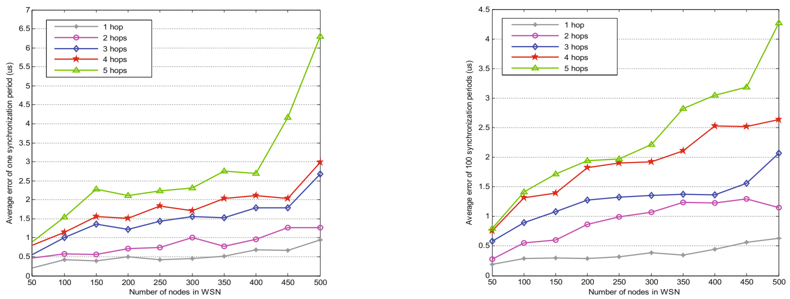
Parameters	Descriptions
Number of nodes in a WSN	50, 100, 150, 200, 250, 300, 350, 400, 450, 500
Number of hops	1, 2, 3, 4, 5
Initial clock value of a node	Random number within 1 (us)
Initial energy value of a node	0.125 J
Accuracy of random delay	0.1 (us)
E_{elec}	50 nJ/bit
E_{amp}	100 pJ/bit/m^2
k	100 bit (synchronization message) 200 bit (ACK message)
Network topology	Random

5.2 Synchronization Accuracy

Through the MATLAB simulations we observed the synchronization accuracy by the average synchronization errors which are calculated after finishing each period of clock synchronization for all nodes in a WSN. That is, after master node receives and updates the complete node information table, the clock value of a node is calculated directly from the table. The average error of a node is calculated by arithmetic square root of all clock values difference of each node with the master node. Similarly, the average synchronization error of nodes in multiple synchronization periods can be analyzed.

Figure 3(a) shows the simulation results of average synchronization errors calculated for a single hop network in one synchronization period, while Fig. 3(b) shows for the 100 synchronization periods. For multi-hop network, we carried out the simulations from single-hop to 5-hops. It can be seen clearly from the figures that, the CGCSP has high synchronization accuracy, even if the WSN reaches 500 sensor nodes. The average synchronization accuracy of nodes in single hop network is less than 1 us, and the average synchronization accuracy is less than 0.5 us for the WSN with less than 300 nodes.

In summary, the influence of network size (number of nodes) and the number of network hops on the synchronization accuracy of CGCSP can be obtained as: (1) With the increase of the number of nodes in network, the synchronization error will gradually increase, which is due to the global synchronization will introduce random errors for each node. However, by increasing the synchronization cycles, the sensor node can constantly update its local clock in a short time, which will achieve better synchronization accuracy. (2) The increase in the number of hops cause the synchronization error to increase because of CGCSP for multi-hop network application, the synchronization error of intermediate node and master node will be transmitted along with the multi-hop network. There is a large accumulation of errors in the broadcast area.



(a) Average synchronization errors in one synchronization period.

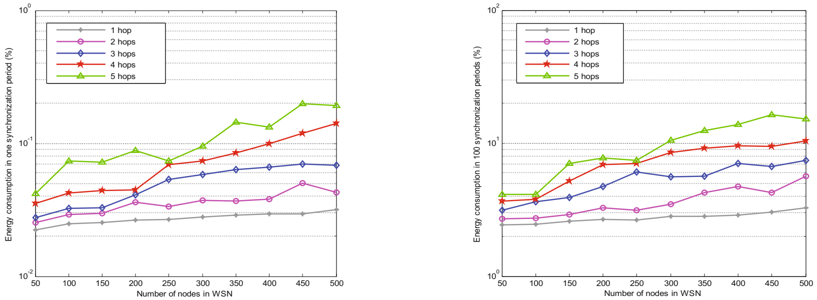
(b) Average synchronization errors in 100 synchronization periods.

Fig. 3. Average synchronization accuracy versus no. of communicating nodes.

5.3 Energy Consumption

The energy efficiency of CGCSP is evaluated by energy consumptions in terms of messages exchange during clock synchronization periods. The energy consumption of a node in synchronization process is estimated according to the Eq. (6). The total energy consumed by the whole network for global clock synchronization is calculated as the percentage of it to its initial energy (energy consumption ratio). The energy consumption of entire network is calculated after the execution of one and more synchronization periods.

Figure 4(a) displays the simulation results of energy consumptions calculated in a single hop network with one synchronization period, while Fig. 4(b) shows that with 100 synchronization periods. For multi-hop network, we carried out the simulation from single-hop network to 5-hops network. It is evident from the figures that, the CGCSP has low synchronization energy consumptions. In the case of completion of one synchronization period, whether single-hop or multi-hop network, the energy consumption ratio is less than 0.1% for WSN with less than 300 nodes, even if the WSN reaches 500 sensor nodes the energy consumption ratio is less than 0.2%.



(a) Energy consumption ratio in one synchronization period.

(b) Energy consumption ratio in 100 synchronization periods.

Fig. 4. Energy efficiency versus no. of communicating nodes.

In summary, the influence of network size (number of nodes) and the number of network hops on the synchronization energy consumptions of CGCSP can be obtained as: (1) With the increase of the number of nodes, the number of information exchange increases, and the energy consumption in the synchronization process is gradually increased. (2) Although better synchronization accuracy can be achieved by increasing the number of synchronization periods, but it will increase the synchronous energy consumptions significantly, so it is necessary to compromise between synchronization accuracy and the energy efficiency. (3) For multi-hop network, with the increase of the number of hops, the synchronous energy consumption increases at a small extent. It is because of communication hops, more hops mean larger spacing of nodes. The synchronization in each hop is achieved by the previous hop of the middle node and not directly by master node and the expected master node for information exchange. Thus, it reduced the information dissemination path and the energy consumption in synchronization did not increase significantly.

Figure 5 shows the results of energy consumption with path loss coefficient ranging from 2–5. It can be seen that with the increase of path loss coefficient, the energy consumption of nodes is increased. When the path loss coefficient is 4, the energy consumption is very large after 10 synchronous periods, and after 100 periods the energy consumption ratio has reached 100%. When the path loss coefficient is 5, the energy consumption in synchronization becomes quite large. Therefore, the CGCSP is more suitable for the indoor WSNs environment.

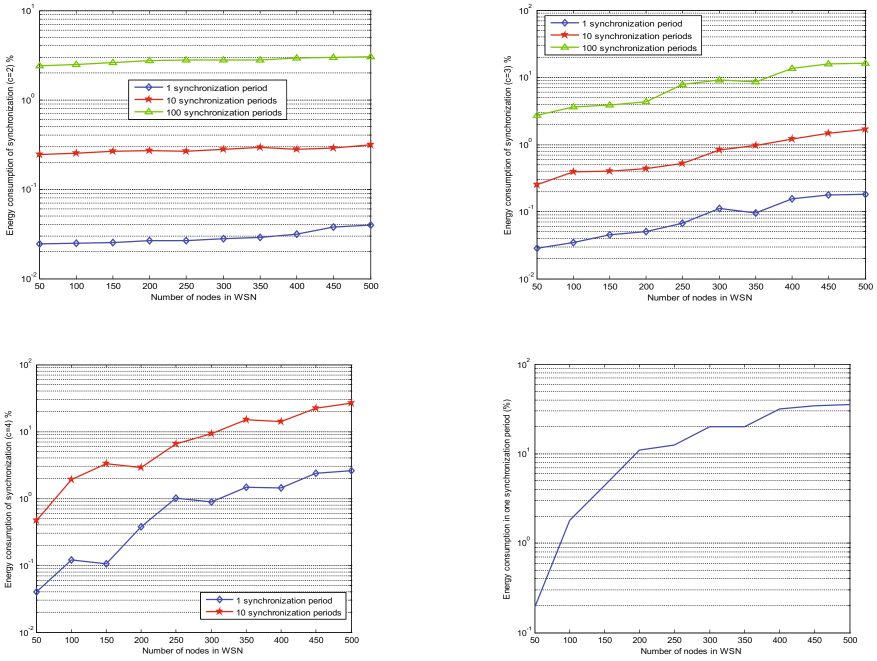
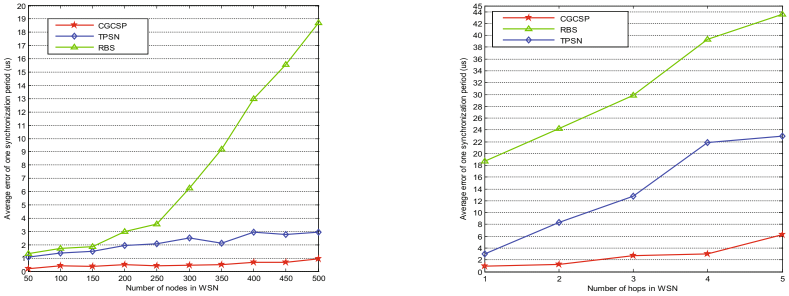


Fig. 5. Energy consumption with different path loss coefficient (c = "2, 3, 4 and 5").

5.4 Performance Comparison

In this section, we compare the performances of the CGCSP with that of the RBS and TPSN, which are two often used typical protocols for WSNs, in single hop network and multi-hop networks respectively with the same simulation parameters.

Figure 6(a) demonstrates that the average synchronization errors of the three algorithms increase with the increase of the network size (the number of nodes). However, the average synchronization error of CGCSP method is obviously lower than that of RBS and TPSN. Considering the case of a large-scale multi-hop WSN with 500 sensor nodes, Fig. 6(b) displays that the average synchronization errors of the three algorithms increase with the increase of the number of hops. However, the average synchronization error of CGCSP is much lower than that of RBS and TPSN in the multi-hop network.



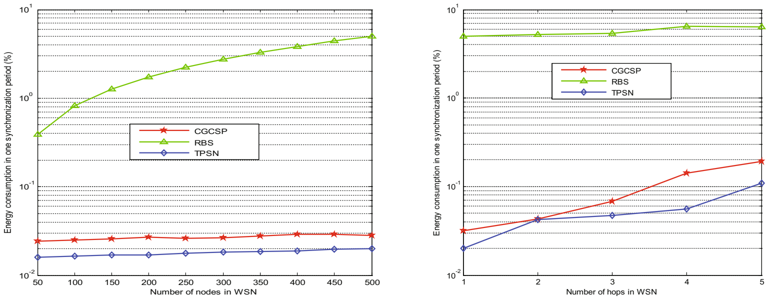
(a) Comparison of the average synchronization error and for single-hop network in one synchronization period.

(b) Comparison of the average synchronization error and number of hops in one synchronization period of a 500-nodes network.

Fig. 6. Average synchronization accuracy comparison.

The results show that the CGCSP achieves the best synchronization accuracy as compared with the other methods. The CGCSP attains least average synchronization errors in case of the same size of network and communication hops.

Figure 7(a) illustrates the simulation results of energy consumption ratios in one synchronization period for a single-hop network, while Fig. 7(b) shows for a 500-nodes multi-hop network. It can be seen that, the RBS algorithm has the maximum synchronization energy consumption due to its large number of information exchanges in synchronization process. While the CGCSP and TPSN has quite low synchronization energy consumptions both in single-hop and multi-hop network. Comparing the CGCSP and TPSN, the energy consumption of CGCSP is slightly higher than that of TPSN. This is due to the hierarchical structure of TPSN algorithm, which makes the nodes can exchange information with adjacent nodes and the information transmission distance can be reduced and as a result the energy consumption is relatively low.



(a) Comparison of the relationships between energy consumption ratio and number of nodes for single-hop network in one synchronization period.

(b) Comparison of the relationships between energy consumption ratio and number of hops in one synchronization period of a 500-nodes network.

Fig. 7. Energy efficiency comparison of three protocols.

6 Conclusions and Future Research Direction

In this paper, we have analyzed the main strategies of the CGCSP, studied its synchronization process and given the procedure of its implementation. By simulation experiments and comparative study, we analyzed the synchronization accuracy and energy efficiency of the CGCSP. The CGCSP employs energy level based cognitive node switch-over mechanism, which can dynamically select master node and expected master node and is reliable. It ensures the stability of global clock synchronization process and keeps a good synchronization accuracy and lower energy consumption under the condition of certain node failure. Simulation results proved the effectiveness of the CGCSP. For the same network size, the CGCSP has higher synchronization accuracy than TPSN and RBS. The proposed method shows lower energy consumption than RBS while has little more overhead than the TPSN.

The research work presented in this paper is accomplished through MATLAB. The study is performed with the parameters exhibited by the physical sensors. As a future research direction a study can be conducted for the implementation of the CGCSP on real sensor networks.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under grant number 61171145.

References

1. Durišić, M.P., Tafa, Z., Dimić, G., Milutinović, V.: A survey of military applications of wireless sensor networks. In: Proceedings of the 1st Mediterranean Conference on Embedded Computing (MECO 2012), pp. 196–199. IEEE (2012)
2. Srbinovska, M., Gavrovski, C., Dimcev, V., Krkoleva, A., Borozan, V.: Environmental parameters monitoring in precision agriculture using wireless sensor networks. J. Clean. Prod. **88**, 297–307 (2015)

3. Ma, J., Wang, H., Yang, D., Cheng, Y.: Challenges: from standards to implementation for industrial wireless sensor networks. Int. J. Distrib. Sens. Netw. **2016**(2016), 1–13 (2016)
4. Li, H., Jia, L., Zhang, Y., Liu, C., Rong, J.: Wireless sensor networks of infrastructure health monitoring for high-speed railway. Shock Vibr. **2016**(2016), 1–11 (2016)
5. Mills, D.: Internet time synchronization: the network time protocol. IEEE Trans. Commun. **39**(10), 1482–1493 (1991)
6. IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE (2008)
7. RFC 5905: Network Time Protocol Version 4: Protocol and Algorithms Specification
8. Li, Q., Rus, D.: Global clock synchronization in sensor networks. IEEE Trans. Comput. **55**(2), 214–226 (2006)
9. Wu, Y.-C., Chaudhari, Q., Serpedin, E.: Clock synchronization of wireless sensor networks. IEEE Signal Process. Mag. **28**(1), 124–138 (2011)
10. Ganeriwal, S., Kumar, R., Sirivastava, M.B.: Timing-sync protocol for sensor networks. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 138–149. ACM (2003)
11. Elson, J., Girod, L., Estrin, D.: Fine-grained network time synchronization using reference broadcasts. ACM SIGOPS Oper. Syst. Rev. **36**(SI), 147–163 (2002)
12. Maroti, M., Kusy, B., Simon, G., Ledeczi, A.: The flooding time synchronization protocol. In: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, pp. 39–49. ACM (2004)
13. Solis, R., Borkar, V.S., Kumar, P.R.: A new distributed time synchronization protocol for multihop wireless networks. In: Proceedings of IEEE Conference on Decision and Control (2006)
14. Hong, Y.-W., Scaglione, A.: A scalable synchronization protocol for large sensor networks and its applications. IEEE J. Spec. Areas Commun. **23**(5), 1085–1099 (2005)
15. Giridhar, A., Kumar, P.R.: Distributed clock synchronization over wireless networks: algorithms and analysis. In: Proceedings of IEEE Conference on Decision and Control (2006)
16. Ahmed, S., Xiao, F., Chen, T.: Achieving relative time synchronization in wireless sensor networks. J. Control Sci. Eng. **2013**, 1–7 (2013)
17. Yanos, S.: Energy aware time synchronization in wireless sensor networks. University of North Texas, December 2006
18. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, pp. 3005–3014 (2000)
19. Tang, X., Xu, J.: Adaptive data collection strategies for lifetime constrained Wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **9**(6), 721–734 (2008)
20. Muhammad, A., Tarek, R.S.: RTSP: an accurate and energy-efficient protocol for clock synchronization in WSNs. IEEE Trans. Instrum. Meas. **62**, 578–589 (2013)

A Generic Web Application Testing and Attack Data Generation Method

Hsiao-Yu Shih[1], Han-Lin Lu[1], Chao-Chun Yeh[1,3], Hsu-Chun Hsiao[4], and Shih-Kun Huang[1,2](✉)

[1] Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan
ncnoa221@gmail.com, littleflyer2015@gmail.com,
skhuang@cs.nctu.edu.tw

[2] Information Technology Service Center, National Chiao Tung University, Hsinchu 300, Taiwan

[3] Computational Intelligence Technology Center, Industrial Technology Research Institute, Hsinchu 300, Taiwan
avainyeh@itri.org.tw

[4] Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan
hchsiao@csie.ntu.edu.tw

Abstract. With the advances of diversified online services, there is an increasing demand for web applications. However, most web applications contain critical bugs affecting their security, allowing unauthorized access and remote code execution. It is challenging for programmers to identify potential vulnerabilities in their applications before releasing the service due to the lack of resources and security knowledge, and thus such hidden defects may remain unnoticed for a long time until being reported by users or third-party risk exposure. In this paper, we develop an automated detection method to support timely and flexible discovery of a wide variety of vulnerability types in web applications. The key insight of our work is adding a lightweight *detecting sensor* that differentiates attack types before performing *symbolic execution*. Based on the technique of symbolic execution, our work generates testing and attack data by tracking the address of program instruction and checking the arguments of dangerous functions. Compared to prior analysis tools that also use symbolic execution, our work flexibly supports the detection of more types of web attacks and improve system flexibility for users thanks to the detecting sensor. We have evaluated our solution by applying this detecting process to several known vulnerabilities on open-source web applications and CTF (Capture The Flag) problems, and detected various types of web attacks successfully.

Keywords: Web application testing · Symbolic execution · Capture The Flag
Software vulnerability

1 Introduction

Web applications have become a significant part of the Web because of the attractive features such as easy installation, customization and high accessibility. However, they are often deployed with critical software bugs that can be maliciously exploited. Once

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 232–247, 2018.
https://doi.org/10.1007/978-3-319-76451-1_22

a weakness is found, it can be exploited to take control of the system. Hence, there is a need for an analysis tool which can automatically detect vulnerabilities and defend against threats.

1.1 Motivation and Objective

Web applications are usually built with multiple utilities for customers in various programming languages. Our previous work, CRAXWeb [1], achieves the goal of detecting XSS and SQL injection attack and generating the corresponding exploit [2–4]. However, many types of attack remain unsupported. We propose to add an attack type differentiator called detecting sensor in a designated call site of the Web service. The complex step of inserting a detecting sensor into the web service components also decreases the flexibility in the testing process. Therefore, we improve CRAXWeb to support the detection of more attack types and speed up the procedure to deploy detecting sensors.

The aim of this work is to extend an existing exploit generator, CRAXWeb, for XSS and SQL injection attack on web applications to implement generic web attack generation. This work is based on a popular dynamic analysis technique in the field of software testing called symbolic execution [5, 6]. Many related works are also based on it. However, most of the works focus on only XSS and SQL injection attack. Our challenge is to protect web applications against multiple types of threats.

1.2 Overview

This paper is organized as follows. Section 2 describes the background of software testing and related web security issues. Section 3 describes and compares related works. Sections 4 and 5 explain our method and implementation, respectively. Experimental results are reported in Sect. 6. Finally, Sect. 7 concludes our paper, with future work.

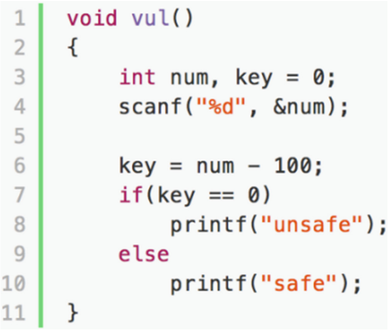
2 Background

2.1 Symbolic Execution

Symbolic execution is a testing approach that executes programs with symbolic rather than concrete inputs. Its objective is to explore as many paths in a program as possible. Before executing, a path constraint is initialized as true. Whenever the program execution encounters an assignment statement that associates with symbolic variable, the symbolic variable will taint other concrete variables as symbolic. When symbolic execution encounters a branch condition, it forks the execution state, following both branch directions and updating the corresponding path constraints on the symbolic input. If the program exits, or terminates unexpectedly, the current path constraint will be solved to compute a concrete test case that drives the program to this execution point.

By considering symbolic execution on the example in Fig. 1, `num` is assigned a symbol X at line 4 since it is user provided data. For the assignment at line 6, the symbolic variable `num` taints the concrete variable `key` and the symbolic value of `key` becomes

X - 100. For the branch at line 7, the execution encounters the symbolic variable `key` and forks a new execution for another new path. One takes the true path with an additional constraint $X - 100 == 0$. The other takes the false path, with an additional constraint $X - 100 \neq 0$. Whenever two forked executions finish, their path constraints can be solved by the constraint solver for generating new test cases. One case is $num = 100$ and another case is $num = 101$(not equal to 100). The process is shown in Figs. 1 and 2



```
 1  void vul()
 2  {
 3      int num, key = 0;
 4      scanf("%d", &num);
 5
 6      key = num - 100;
 7      if(key == 0)
 8          printf("unsafe");
 9      else
10          printf("safe");
11  }
```

Fig. 1. Sample code

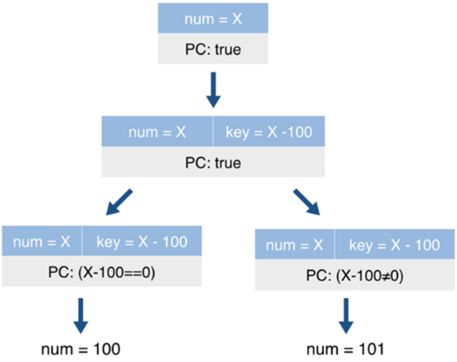


Fig. 2. Symbolic execution for sample code

2.2 Web Security Issues

Cross Site Scripting (XSS)

Cross Site Scripting (XSS) is a common attack vector that injects code into website to complete a range of actions from stealing logs to installing malicious software on user's computer. Two primary types of XSS are reflected XSS and stored XSS. In reflected XSS, an attacker crafts a URL containing a malicious string and sends it to the victim. Whoever clicks the link is going to have the script execute in the browser. In stored XSS, also known as persistent XSS, an attacker injects the malicious payload into databases or visitor logs and has it be available to all visiting users, the payload will run in each of the victims' browsers.

Cross-Site Request Forgery (CSRF)

Cross-site request forgery (CSRF) is a type of website exploit caused by transmitting unauthorized commands from a user whom the web application trusts. As opposed to XSS, which exploits the user's trust for a website, CSRF exploits a website's trust for a particular user's browser. CSRF attack forces a logged-on victim's browser to send a forged HTTP request, which allows the attacker to launch any desired requests against the website, without the website being able to distinguish whether the requests are legitimate or not.

SQL Injection

An SQL injection vulnerability occurs when the data from an untrusted source are used to dynamically construct a SQL query. Attackers trick the system by executing malicious

SQL commands to manipulate the backend database [7]. An SQL injection attack may result in data theft, loss, modification or corruption, or even complete takeover of the server.

Command Injection

Command injection occurs when an application passes unsafe user supplied data to a system shell. An attacker can execute operating system commands with the privileges of the vulnerable application. There are many ways to exploit a command injection, such as by injecting the command inside backticks (for example `'ls'`) and running another command if the first one succeeds (for example `&&ls`).

File Inclusion

A file inclusion vulnerability occurs when a user-controlled parameter is used as part of a file name in a call to an including function (`require()`, or `include()` in PHP for example). Depending on whether the file is remote or local, file inclusion can be categorized into Remote File Inclusion (RFI) and Local File Inclusion (LFI).

Remote File Inclusion allows an attack to include and execute a remotely hosted file. Since the included file is controllable, the attack can run arbitrary code either on the client side or on the server. In the scenario of Local File Inclusion, the attacker can access unauthorized files or utilize directory traversal characters to retrieve sensitive files available in other directories.

3 Related Work

This section presents a comprehensive survey of previous work undertaken in the field of testing and vulnerability analysis for web scripting languages. There are three main topics: Symbolic Execution based Test Generation, Static/Dynamic Analysis based Attack Detection, and Symbolic Execution based Attack Detection. Symbolic Execution based Test Generation.

Apollo [8] is a tool that uses concrete and symbolic (concolic) execution to generate failure-inducing inputs for PHP web applications. *Jalangi* [9] is a dynamic analysis framework for JavaScript that applies concolic execution to generate function arguments. *SymJS* [10] contains a symbolic execution engine for JavaScript and an automatic event explorer. *Jalangi* works for pure JavaScript programs, while *SymJS* works for general web applications. *Derailer* [11] is a security bugs finding tool for Ruby on Rails web applications. *Chef* [12] is a symbolic execution engines relying on the S^{2}E [13] for Python and Lua analysis. *MultiSE* [14] extends *Jalangi* and introduces a new technique for merging states to improve its effectiveness.

3.1 Static/Dynamic Analysis Based Attack Detection

Pixy [15] performs interprocedural flow-sensitive data flow analysis to detect XSS vulnerabilities in PHP scripts. *XSS-GUARD* [16] dynamically learns the set of scripts that a web application intends to create for any HTML request. *PIUIVT* [17] enhances the efficiency of invalid test inputs generation depending on feedback of analysis.

MySQLInjectior [18] is a web scanning tool to detect SQL injection vulnerabilities based on the identified styles. *NKSI Scan* [19] is a model based penetration test method for the SQL injection vulnerabilities. It can generate test case covering different types and patterns of SQL injection attack input. *Zheng et al.* [20] propose a path- and context-sensitive interprocedural analysis to detect remote code execution vulnerabilities on PHP applications. *XSSDM* [21] proposes a context-sensitive approach based on static taint analysis and pattern matching techniques to detect XSS vulnerabilities. *Joza* [22] is a hybrid approach which combines advantages of negative [23] and positive inference [24] for SQL injection detection. *DEKANT* [25] uses static analysis with the ability to learn to characterize vulnerabilities based on annotated source code slices.

3.2 Symbolic Execution Based Attack Detection

SAFELI [26] inspects Java to automatically generate SQL injection attack scenarios. *Adrilla* [27] is an exploit generator which stems from Apollo. It combines concolic testing and dynamic taint analysis to generate concrete attack vectors for PHP web applications. *Kudzu* [28] is the first symbolic execution based framework for JavaScript code analysis. It uses attack grammars to solve the exploit and finally finds out two unknown vulnerabilities. *Rubyx* [29] is a symbolic executor for Ruby, with builtin support for specification and verification of scripts. It proves complex security and correctness properties of Ruby-on-Rails web applications. *Huang et al.* [30] proposed a hybrid vulnerability analyzer for Java that applies symbolic execution to generate path constraints. *Codeminer* [31] combines static analysis and symbolic code execution to identify XSS and SQL injection vulnerabilities on PHP web applications. Compared to these works, our framework can detect XSS and SQL injection attack for the web applications written in any language. Moreover, for PHP web applications, our framework detects more types of attacks such as command injection, code injection, and file inclusion.

4 Method

Our work is based on the Selective Symbolic Execution (S^{2}E) [13] framework, which supports application emulation using QEMU. Figure 3 is the model of our method, which is divided into four main parts: Symbolic Environment, Dangerous Function Analysis, Symbolic Argument Checking, and Host Management.

The Guest OS comprises a client and a server which runs one or more back-end web applications and a database, working like the real-world web service environment. The only difference is the way client communicates with server. Client sends symbolic data to server along with HTTP requests. The Host OS keeps track of the program counter during the request processing. When the addresses of dangerous functions are reached, it will check whether the arguments of the functions are symbolic. The Host Management contains a list of functions we are interested in and a configuration file used to control the symbolic execution in the Host OS.

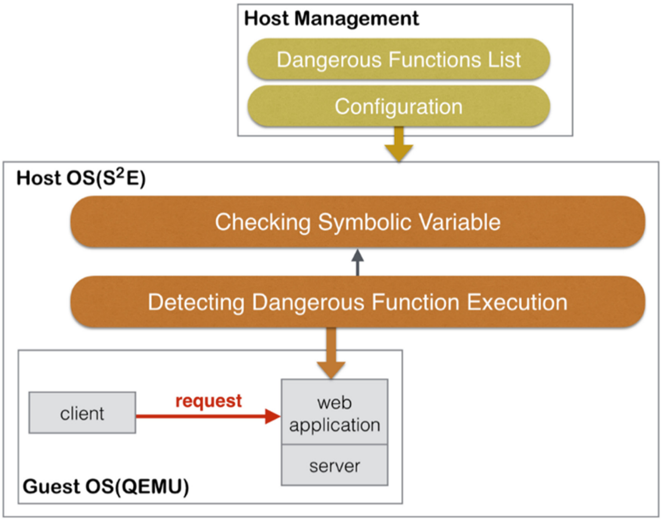


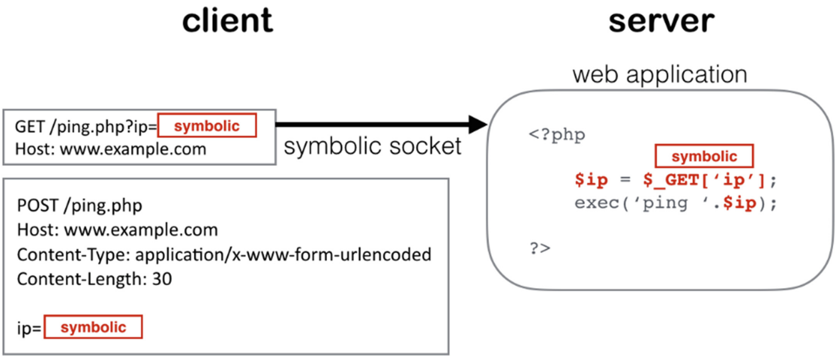
Fig. 3. Symbolic environment, dangerous function analysis, symbolic argument checking, and host management

4.1 Symbolic Environment

Symbolic Socket

To attack web applications, an attacker inject unexpected inputs to invoke abnormal program execution and reaction, such as system crash and sensitive data leakage. These malicious inputs are propagated within HTTP request, in the form of GET parameters in URL or POST data in message body. For SQL injection, the single quotation mark and the UNION operator are commonly leveraged to craft a malicious query, which is joined into the original query intended to be run by the web application. For command injection, the crafted command which is used as the arguments of dangerous functions will be processed on the operating system.

In order to explore all possible paths through the whole web executing procedure from request to response that correspond to all possible inputs, we have to make these inputs symbolic. Hence, we adopt symbolic socket, which is composed of HTTP request and symbolic data, to act as the communicator between the server and the client. The symbolic data is injected into HTTP request to replace the value of original inputs from users and passed to the web server along with the HTTP request. If the symbolic data can reach the functions we are interested in, it implies that the arguments of these functions can be controlled by the original symbolic data. Figure 4 shows the propagation of symbolic data.



```
GET /ping.php?ip=symbolic
Host: www.example.com
```

```
POST /ping.php
Host: www.example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 30

ip=symbolic
```

```
<?php

    $ip = $_GET['ip'];
    exec('ping '.$ip);

?>
```

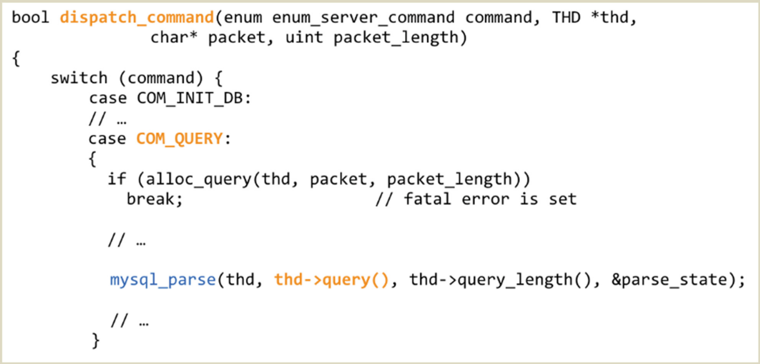
Fig. 4. Symbolic data propagates to the server along with HTTP request

4.2 Dangerous Function Analysis

Target Function Detection

To facilitate web security testing, our technique should be able to detect as many types of vulnerabilities as possible. Because most of the vulnerabilities occur when untrusted data is passed into and executed by functions in web application languages, what we care about in this detecting method is the address of such dangerous functions. S^{2}E is extensible to support different architectures and features by means of a plugin interface. We implement a customized plugin in the Host OS to monitor the address that the symbolic data flows through the programs in the Guest OS during the symbolic execution.

In order to fulfill the goal of detecting SQL injection attack in web applications which are written in different programming languages, it is essential to analyze the query processing in the database server. The `dispatch_command()` function in MySQL source code is where MySQL actually starts the analysis of commands, including queries, prepared statements and other command types. Figure 5 shows part of the code in `dispatch_command()`. As the name of the function implies, it is responsible for dispatching the query to the appropriate handler. Since the SQL query run by a web



```
bool dispatch_command(enum enum_server_command command, THD *thd,
              char* packet, uint packet_length)
{
    switch (command) {
        case COM_INIT_DB:
        // …
        case COM_QUERY:
        {
          if (alloc_query(thd, packet, packet_length))
            break;                    // fatal error is set

          // …

          mysql_parse(thd, thd->query(), thd->query_length(), &parse_state);

          // …
        }
```

Fig. 5. MySQL function dispatch_command()

application is a standard SQL query over the connection, we focus on the *COM_QUERY* block in the switch statement. If the block is reached during symbolic execution, we can continuingly check whether the query string (i.e. *thd-> query()*) is controllable.

For other types of attack such as command injection, code injection, file inclusion and more, we target at the functions in PHP since it is the most widely used web application programming language and has raised substantial number of security issues due to the improper use of functions. Take command injection for example, one of our target functions is `shell_exec()`, which allows users to execute an external program. Figure 6 shows the `shell_exec()` function in *ext/standard/exec.c* of PHP 5.5.38. The char pointer, *char *command*, is the command that will be executed as well as the argument that we have to check.

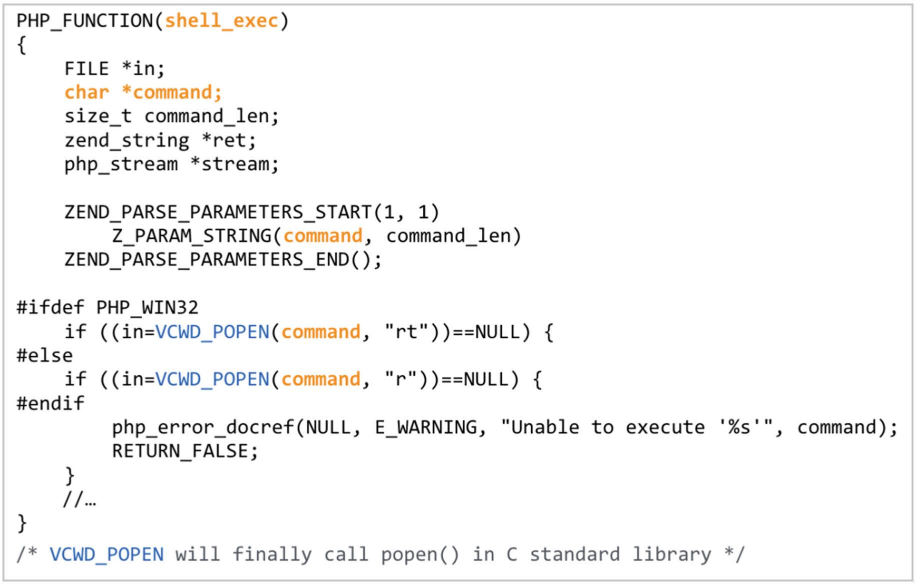
Symbolic Argument Checking

If the recorded functions are reached, we have to continuingly verify the controllability of the arguments. S^{2}E fetches blocks of guest code, translates them to the host's instruction set, and passes the resulting translation to the execution engine. It determines which code to fetch and translate by reading the state of the virtual CPU and the guest memory.

Function Argument Checking.

It is time-consuming to insert s2e opcode to PHP or MySQL source code and recompile the binaries.

To deal with this problem, we design a method to analyze the program state at the Host OS when S^{2}E notices that a dangerous function is executed instead of invoking the checking process from the Guest OS. This reduces the manipulation on web service components, leaving the web environment easy to deploy. The web server, database,



```
PHP_FUNCTION(shell_exec)
{
    FILE *in;
    char *command;
    size_t command_len;
    zend_string *ret;
    php_stream *stream;

    ZEND_PARSE_PARAMETERS_START(1, 1)
        Z_PARAM_STRING(command, command_len)
    ZEND_PARSE_PARAMETERS_END();

#ifdef PHP_WIN32
    if ((in=VCWD_POPEN(command, "rt"))==NULL) {
#else
    if ((in=VCWD_POPEN(command, "r"))==NULL) {
#endif
        php_error_docref(NULL, E_WARNING, "Unable to execute '%s'", command);
        RETURN_FALSE;
    }
    //…
}
/* VCWD_POPEN will finally call popen() in C standard library */
```

Fig. 6. PHP function shell_exec()

and programming language can be built through a package manager like pip or apt-get without hardcoding other functions needed by symbolic execution.

The method to analysis function arguments is divided into two steps: reading the memory address from the register where the function argument stores and determining whether the value in the memory is symbolic or concrete. Figure 7 shows an overview of the workflow.

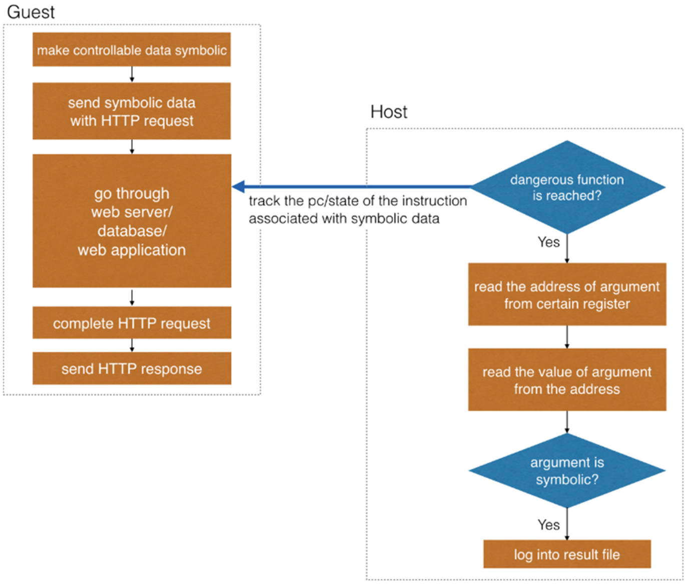
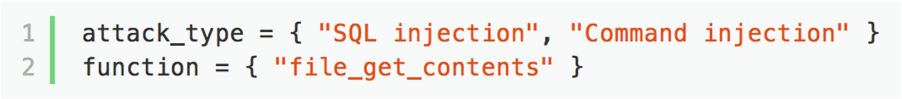

Guest
make controllable data symbolic
send symbolic data with HTTP request
go through web server/ database/ web application
complete HTTP request
send HTTP response
Host
track the pc/state of the instruction associated with symbolic data
dangerous function is reached?
Yes
read the address of argument from certain register
read the value of argument from the address
argument is symbolic?
Yes
log into result file

Fig. 7. The overview of the workflow

4.3 Host Management

Dangerous Function List

In order to reduce the complexity of processes for users, we build a list of functions and their corresponding attack types. Users can choose what kind of detection they want to implement on the web application by writing a python script to specify the attack types or functions that are of interest. There is no need for users to consider the addresses of the function and the argument. Figure 8 is the example of the python script. In this example, although the `file_get_contents()` function does not belong to any of the two attack types, it is also the target function.



```
attack_type = { "SQL injection", "Command injection" }
function = { "file_get_contents" }
```

Fig. 8. Python script example that is provided by user

5 Implementation

In this section, we explain in detail how our method is implemented on S^{2}E.The first part is Symbolic Environment including the whole system architecture of our framework and the propagation of symbolic data. The second part is Host OS Risk Detection; it relates to the plugin we design to detect function execution and identify symbolic arguments. The third part is Host Management; it works as the communicator between users and S^{2}E.

5.1 Symbolic Environment

System Architecture

The architecture of the system is summarized in Fig. 9: the web application is built in the Guest OS, which is running on a machine emulator called QEMU. The Host OS constructed by S^{2}E receives the information propagated from the Guest OS, and then our customized S^{2}E plugin can verify whether the dangerous functions are reached and whether the function arguments are symbolic. Users can specify the functions or attack types in a python script. The configuration writer creates the configuration file depending on the script and the dangerous function list to control the plugin.

Most web applications are based on the client-server architecture where the client submits data while the server stores and retrieves data. We make the testing application run on top of Debian 7 with Apache as the web server and MySQL as the database in light of flexibility and accessibility of use.

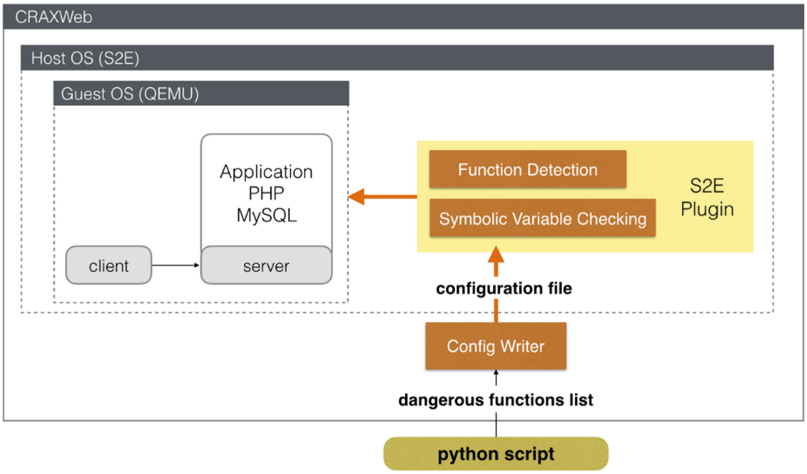
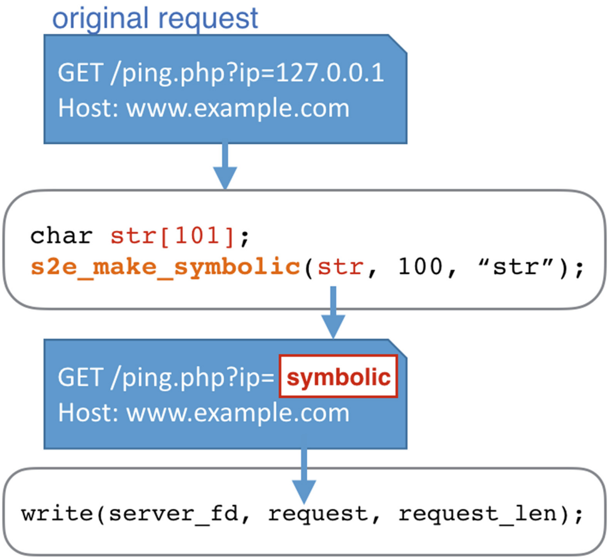


Fig. 9. System Architecture

Symbolic Socket

We deploy a program acting as the client in the Guest OS to generate symbolic socket, which is made up of a HTTP request with injected symbolic data. Then, the HTTP request message with the symbolic string is written to the socket and sent to the web server, as shown in Fig. 10.



```
GET /ping.php?ip=127.0.0.1
Host: www.example.com
```

```
char str[101];
s2e_make_symbolic(str, 100, “str”);
```

```
GET /ping.php?ip= symbolic
Host: www.example.com
```

```
write(server_fd, request, request_len);
```

Fig. 10. Generate a symbolic socket

Whenever a branch referring to symbolic data is encountered, the entire states (i.e. memory, registers and PC) will be forked and each side of the branch will be explored by S^{2}E.

5.2 Dangerous Function Analysis

In order to detect multiple types of attacks, we focus on the execution of the functions that might be used by attackers with malicious intentions. In this section, we detail the analysis process of finding function address and the implementation of checking symbolic variable in our customized plugin.

Target Function Detection

- **MySQL Executable Analysis**

MySQL supplies different executables to serve different purposes for users. For example, *mysql* is a command-line client for executing SQL statement interactively and *mysqld* is the server executable. Since the symbolic data is sent to the web server along with HTTP request from the client-side, the target to analyze is *mysqld*, which is the server daemon in the Unix-like operating system.

By reversing *mysqld* with the IDA Pro Disassembler, we can find the address of `dispatch_command()` function, which is the entry point of MySQL query.

In addition to the address of the function, the value of the argument is also necessary for checking symbolic variable. S^2E keeps track of the instruction executed in the Guest OS with the corresponding state. We observe the disassembly code of `dispatch_command()` and find the register where the argument is stored.

- **PHP Module Analysis**

Using the PHP module to execute PHP scripts on Apache is the default mode set at the creating phase of most web frameworks. The PHP module acts as the PHP interpreter that is embedded in each Apache processes, which means that no external PHP process is required. As the interpreter is started along with Apache, it can cache certain information and need not to repeat the same tasks each time a script is executed, leading to the good performance on PHP heavy sites.

Apache loads numerous modules when the service starts, including the PHP module which is named *libphp5.so*. In order to get the base address of the PHP module inside the Apache process, we use *pmap* command in Linux which can report memory map of a process to list the information related to *libphp5.so*.

Then we reverse *libphp5.so* in the Apache process to get the starting address and length of functions such as zif_shell_exec() or shell_exec() in PHP.

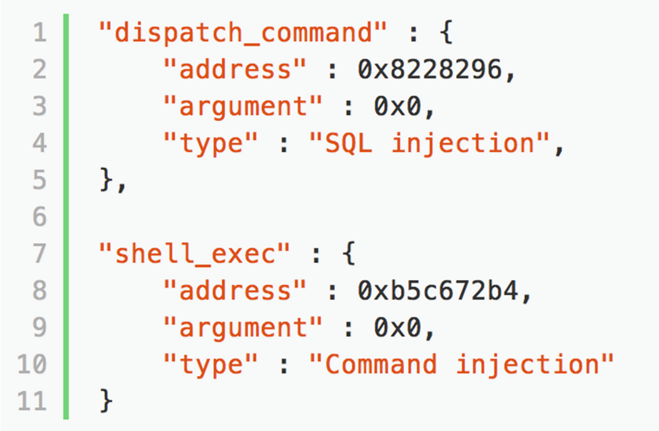
Symbolic Argument Checking

S^2E provides macros to access the registers and memory of S^2E-specific state. To get the contents in registers, we use `readCpuRegisterConcrete()` macro to read concrete value from general purpose CPU register. There are two macros to read the content in memory according to the status of the target memory: `readMemoryConcrete()` gets concrete data, if the target memory is concrete status; `readMemory8()` gets symbolic data, if the target memory is symbolic status. However, `readMemoryConcrete()` fails if the value is symbolic. Since the argument has chance to be symbolic, we use `readMemory8()` to read the content from memory and then cast it into the `KLEE Expression`. S^2E uses the `KLEE Expression` class as the fundamental building block of all values in the emulated memory object. A concrete value is merely a `KLEE ConstantExpression` which is derived from `KLEE Expression`. We determine a value is symbolic if its cast expression is not a `KLEE ConstantExpression`.

5.3 Host Management

Dangerous Function List

The dangerous function list is written in JSON format. It contains numerous function names as the keys and each of them has the following components: the function address, the offset of the register which stores the function argument, and the attack type. S^2E uses the predefined offset value to access each register from QEMU. Figure 11 is the example of the function list.



```
"dispatch_command" : {
    "address" : 0x8228296,
    "argument" : 0x0,
    "type" : "SQL injection",
},

"shell_exec" : {
    "address" : 0xb5c672b4,
    "argument" : 0x0,
    "type" : "Command injection"
}
```

Fig. 11. The example of the function list

6 Evaluation

6.1 Evaluation of Vulnerable Applications

Experimental Environment

All experiments performed on a host hardware including a 2.4 Ghz CPU with 8 cores, 8 GB physical memory and host OS with Ubuntu 12.04 64-bit desktop edition. The guest environment that is emulated by QEMU includes 2.8 GHz CPU with a single core, 128 MB physical memory and guest OS with Debian 7 32-bit for Linux platform. The software environment is based on S^{2}E 1.0. The database handler is based on MySQL 5.5.49 and the PHP version is 5.5.38.

Experimental Results

The experiment reports the vulnerabilities detection on different platforms to prove the feasibility of platform-independent web testing with our method. Test 1 is a PHP web service that contains numbers of dangerous functions. Test 2 is a website with SQL injection vulnerability and it is built on a Python web framework called Flask. mfw is a challenge of CSAW online CTF in 2016. The forth test case is the web services of RCTF final attack-and-defense contests in 2015; it is built on Codeigniter and with various types of vulnerabilities. The fifth, sixth, and seventh test cases are both the plugin of Wordpress and have been recorded in the CVE list. Table 1 shows the experimental result of vulnerability detection.

Table 1. Evaluation of vulnerable applications

Test case	Attack types	Detected functions	# of lines	Testing time (sec)	Platform	CVE
Test 1	SQLi, Commandi, LFI	mysql_query, system, shell_exec, assert, fopen	55	102.06	PHP	
Test 2	SQLi	MySQL–dispatch_commend	36	31.66	Python (Flask)	
Test 3	Code injection	assert	62	6.25	PHP	
Test 4	SQLi, Code injection	create_function, unserialize	44553	34.59/41.7	PHP (Codeigniter)	
Test 5	Commandi	shell_exec	23086	33.53	PHP	2015-5227
Test 6	Code injection	call_user_func	5377	60.19	PHP	2014-1215
Test 7	Path traversal	file_get_contents	2264	48.58	PHP	2014-5368

*Test 3: mfw (CSAW CTF 2016 web 125), Test 4: RCTF Final 2015, Test 5: Landing Pages (WordPress plugin), Test 6: Download Manager (WordPress plugin), Test 7: wp-source-control (WordPress plugin)

7 Conclusion and Future Work

The aim of this work is to extend an existing dynamic analysis framework to implement automatic attack detection for web framework. By detecting the execution of dangerous functions, developers can figure out potential vulnerabilities before releasing the web service. This means that software flaws can be fixed early on, and that developers can complete quick security audits.

Our work fulfills the goal of multiple types of web attacks, and has implemented the testing procedure on web applications that are built on different framework and written in different programming languages. The experimental result proved the feasibility of our implementation. In addition, some of the test cases were announced as known vulnerabilities in the CVE database.

Our work can automatically detect SQL injection and XSS attack and generate corresponding exploit string. A SQL injection exploit payload possibly contains the string such as “*’or 1 = 1;–*”, so we set the exploit generator to solve the constraints to generate an input that formed by the basic exploit string. Thanks to our generic construction, it is also possible to generate exploits for other types of web security issues with the same method. By considering the exploit generation on code injection, when the symbolic data reaches the `eval()` or `assert()` functions, exploit generator can continuingly generate the exploit string that formed by “`system(‘ls’)`”.

Acknowledgements. This work was supported by the Institute for Information Industry under the grant 106-EC-17-D-11-1502.

References

1. Huang, S.-K., Lu, H.-L., Leong, W.-M., Liu, H.: Craxweb: automatic web application testing and attack generation. In: 2013 IEEE 7th International Conference on Software Security and Reliability (SERE), pp. 208–217. IEEE (2013)
2. Bisht, P., Hinrichs, T., Skrupsky, N., Venkatakrishnan, V.: WAPTEC: whitebox analysis of web applications for parameter tampering exploit construction. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 575–586. ACM (2011)
3. Martin, M., Lam, M.S.: Automatic generation of XSS and SQL injection attacks with goal-directed model checking. In: Proceedings of the 17th Conference on Security Symposium, pp. 31–43. USENIX Association (2008)
4. Avgerinos, T., Cha, S.K., Rebert, A., Schwartz, E.J., Woo, M., Brumley, D.: Automatic exploit generation. Commun. ACM **57**(2), 74–84 (2014)
5. King, J.C.: Symbolic execution and program testing. Commun. ACM **19**(7), 385–394 (1976)
6. Schwartz, E.J., Avgerinos, T., Brumley, D.: All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask). In: 2010 IEEE Symposium on Security and Privacy (SP), pp. 317–331. IEEE (2010)
7. Halfond, W.G., Viegas, J., Orso, A.: A classification of SQL-injection attacks and countermeasures. In: Proceedings of the IEEE International Symposium on Secure Software Engineering, vol. 1, pp. 13–15. IEEE (2006)
8. Artzi, S., et al.: Finding bugs in dynamic web applications. In: Proceedings of the 2008 International Symposium on Software Testing and Analysis, pp. 261–272. ACM (2008)
9. Sen, K., Kalasapur, S., Brutch, T., Gibbs, S.: Jalangi: a selective record-replay and dynamic analysis framework for JavaScript. In: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, pp. 488–498. ACM (2013)
10. Li, G., Andreasen, E., Ghosh, I.: SymJS: automatic symbolic testing of JavaScript web applications. In: Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 449–459. ACM (2014)
11. Near, J.P., Jackson, D.: Derailer: interactive security analysis for web applications. In: Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, pp. 587–598. ACM (2014)
12. Bucur, S., Kinder, J., Candea, G.: Prototyping symbolic execution engines for interpreted languages. ACM SIGARCH Comput. Archit. News **42**(1), 239–254 (2014)
13. Chipounov, V., Kuznetsov, V., Candea, G.: S2E: a platform for in-vivo multi-path analysis of software systems. ACM SIGPLAN Not. **46**(3), 265–278 (2011)
14. Sen, K., Necula, G., Gong, L., Choi, W.: MultiSE: multi-path symbolic execution using value summaries. In: Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, pp. 842–853. ACM (2015)
15. Jovanovic, N., Kruegel, C., Kirda, E.: Pixy: a static analysis tool for detecting web application vulnerabilities. In: 2006 IEEE Symposium on Security and Privacy, pp. 258–263. IEEE (2006)
16. Bisht, P., Venkatakrishnan, V.: XSS-GUARD: precise dynamic prevention of cross-site scripting attacks. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 23–43. Springer (2008)
17. Li, N., Xie, T., Jin, M., Liu, C.: Perturbation-based user-input-validation testing of web applications. J. Syst. Softw. **83**(11), 2263–2274 (2010)
18. Ali, A.B.M., Abdullah, M.S., Alostad, J.: SQL-injection vulnerability scanning tool for automatic creation of SQL-injection attacks. Procedia Comput. Sci. **3**, 453–458 (2011)

19. Tian, W., Yang, J.-F., Xu, J., Si, G.-N.: Attack model based penetration test for SQL injection vulnerability. In: 2012 IEEE 36th Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 589–594. IEEE (2012)
20. Zheng, Y., Zhang, X.: Path sensitive static analysis of web applications for remote code execution vulnerability detection. In: Proceedings of the 2013 International Conference on Software Engineering, pp. 652–661. IEEE Press (2013)
21. Gupta, M.K., Govil, M.C., Singh, G., Sharma, P., XSSDM: towards detection and mitigation of cross-site scripting vulnerabilities in web applications. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2010–2015. IEEE (2015)
22. Naderi-Afooshteh, A., Nguyen-Tuong, A., Bagheri-Marzijarani, M., Hiser, J.D., Davidson, J.W.: Joza: hybrid taint inference for defeating web application SQL injection attacks. In: 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 172–183. IEEE (2015)
23. Sekar, R.: An efficient black-box technique for defeating web application attacks. In: NDSS (2009)
24. Nguyen-Tuong, A., et al.: To B or not to B: blessing OS commands with software DNA shotgun sequencing. In: 2014 Tenth European Dependable Computing Conference (EDCC), pp. 238–249. IEEE (2014)
25. Medeiros, I., Neves, N., Correia, M.: DEKANT: a static analysis tool that learns to detect web application vulnerabilities. In: Proceedings of the 25th International Symposium on Software Testing and Analysis, pp. 1–11. ACM (2016)
26. Fu, X., Qian, K.: SAFELI: SQL injection scanner using symbolic execution. In: Proceedings of the 2008 Workshop on Testing, Analysis, and Verification of Web Services and Applications, pp. 34–39. ACM (2008)
27. Kieyzun, A., Guo, P.J., Jayaraman, K., Ernst, M.D.: Automatic creation of SQL injection and cross-site scripting attacks. In: IEEE 31st International Conference on Software Engineering, ICSE 2009, pp. 199–209. IEEE (2009)
28. Saxena, P., Akhawe, D., Hanna, S., Mao, F., McCamant, S., Song, D.: A symbolic execution framework for javascript. In: 2010 IEEE Symposium on Security and Privacy (SP), pp. 513–528. IEEE (2010)
29. Chaudhuri, A., Foster, J.S.: Symbolic security analysis of ruby-on-rails web applications. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 585–594. ACM (2010)
30. Huang, Y.-Y., Chen, K., Chiang, S.-L.: Finding security vulnerabilities in Java Web applications with test generation and dynamic taint analysis. In: Proceedings of the 2011 2nd International Congress on Computer Applications and Computational Science, pp. 133–138. Springer (2012)
31. Agosta, G., Barenghi, A., Parata, A., Pelosi, G.: Automated security analysis of dynamic web applications through symbolic code execution. In: 2012 Ninth International Conference on Information Technology: New Generations (ITNG), pp. 189–194. IEEE (2012)

The Study of Improvement and Risk Evaluation for Mobile Application Security Testing

Huey-Yeh Lin, Hung-Chang Chang(✉), and Yung-Chuan Su

Department of Finance, National Formosa University, Yunlin, Taiwan
linhykoo@gmail.com, alexchiang@iis.sinica.edu.tw

Abstract. The popularity of mobile devices has caused them to become indispensable to, and because of increasing dependency on mobile devices following sharp growth in mobile device applications, effective security testing specifications have become essential. However, developers do not prioritize security during mobile application development, causing unscrupulous individuals to exploit loopholes or vulnerabilities in the applications or develop malicious applications to steal sensitive user data, resulting in user information leakage and financial losses. The security specifications for mobile device applications in Taiwan regarding data authorization, data storage, data protection, transmission protocol, transmission protection, application execution, application security, system execution, and system security remain inadequate. Mobile device testing specifications were analyzed in this study, and the specification priorities of documents across countries were categorized. The Open Web Application Security Project and National Institute of Standards and Technology were used as the specification standard with the Cloud Security Alliance's white paper on mobile device specifications to provide more complete security testing specifications for mobile applications. Recommendations were provided based on the testing procedures, improvement methods, and risk assessment of the test items to reduce personal information leakage and financial losses.

Keywords: Mobile phone security · Mobile applications
Inspection specification

1 Introduction

International organizations, such as the Open Web Application Security Project (OWASP), U.S. National Institute of Standards and Technology (NIST), and Chinese Ministry of Industry and Information Technology, and domestic organizations, such as the Industrial Development Bureau of the Taiwanese Ministry of Economic Affairs (in particular, its app inspection specifications version 2.0), National Communications Commission, and Cloud Security Alliance (CSA), have each formulated their own mobile device security specifications. Domestic inspection specifications on data authorization, data storage, data protection, transmission protocols, transmission protection, application execution, application security, system implementation, and system security specifications are inadequate, and security concerns have been raised regarding the specifications for application and system security inspections. This study improved the

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 248–256, 2018.
https://doi.org/10.1007/978-3-319-76451-1_23

domestic inspection specifications for mobile applications to allow examiners to assess whether the investigated mobile devices pose security concerns by using the inspection items and risk levels. The inspection items in the improved mobile application specifications help developers and users track mobile device security before launching applications and help developers consult the risk assessment results and seek recommendations.

2 Literature Review

2.1 Literature Review

The literature review covered numerous strategies with distinct priorities, objectives, and perspectives and focused on research findings, methods, theories, and applications. It also integrated ideas and practices from other studies, assessed the literature, established links between relevant fields, and identified the central topics in particular fields. The review of theories describes crucial proposed or conducted experiments and how abstract concepts from different theories were reorganized and reintegrated (Cooper 1998). In addition to theories, academic discourse, and works in relevant fields, the review references online information, annual reports, relevant record documents, operational specifications, and domestic mobile device security specifications published by regulatory agencies responsible for mobile device security to provide a clearer concept and reference for developing inspection specifications for mobile application security.

2.2 OWASP

OWASP is an international organization that works to establish OWASP foundations globally. It is committed to assisting corporations to design, develop, acquire, operate, and maintain secure applications. All information on the tools, documentation, forums, and chapters used by OWASP is provided free of charge to those interested in application security and its improvement [1].

2.3 NIST

The NIST establishes a stable foundation in physics, biology, and engineering. It provides standards, reference data, and services for applied research, measurement technology, and studies on test methods, and enjoys a high reputation in the international community [2].

2.4 Industrial Development Bureau, Ministry of Economic Affairs

The Industrial Development Bureau provides complete industrial services to meet the demand of industries to facilitate Taiwan's industrial upgrading and transformation, counseling firms to strengthen their operations and improving the industry's productivity and international competitiveness. It assists enterprises in responding to changes in the industrial environment by supporting industrial policy development and strategy formulation and developing projects that promote industrial transformation. The bureau also helps develop and manage industrial districts and draft fiscal and financial measures on industrial development. It additionally provides safety counseling and factory management to control industrial pollution [3].

2.5 NCC

The NCC formulates specifications to ensure effective competition in the telecommunications industry, which has a key role in national industrial development. To cope with the trends in innovation associated with global digital convergence and integrate the existing rights of communication and dispersion, the government resolved to establish a supervisory authority to oversee the integration of telecommunications information and dissemination [4].

2.6 CSA

The CSA is a global nonprofit organization, and the CSA Taiwan Chapter was established in 2011, whereas a civil society organization was formed in the Ministry of the Interior in 2015. It has since promoted emerging information security topics, such as cloud service security, in Taiwan. To improve the trust relationship between cloud services in Taiwan and their users, a fair and objective standard was established, combining the global cloud service security certification program CSA STAR and third-party authentication [5].

2.7 Risk Assessment

The three main elements of information security, known as the CIA triad of information security, are as follows [6]:

- Confidentiality: data, during transfer and storage, should be inaccessible to unauthorized users.
- Integrity: unauthorized users should be prevented from tampering with data during data transfer and storage.
- Availability: mobile device resources should remain available.

The ISO/IEC 27000:2009 and ISO 31000:2009 series of the international information security management standard provide the principles and guidelines for risk management. Through its structured and systemic approach, the various intangible uncertainties that are difficult to describe are managed more transparently.

3 Research Method Steps and Analysis

3.1 Mobile Application Specifications

The research method in this study was to apply the international specifications, such as the OWASP's top 10 vulnerabilities, OWASP Mobile Security Guide, NIST SP800-163.164, ITU-T YD/T 2407, and the CSA's Mobile Application Security Testing (MAST) initiative as the specification for the inspected mobile applications items. To understand the target directions each international organization's specifications, the specification content was defined in accordance with data access security, transmission protocol security, application execution security, and system execution security to determine the expected goal of the document specification. The OWASP's top 10 weaknesses include data access security, transmission protocol security, application execution security, and system execution security as priorities. NIST SP800-163 includes specifications regarding data access security and system execution security but neglects application execution security and system execution security. NIST SP800-164 includes specifications regarding data access security, and it differs from NIST SP800-163 by regulating system execution security but neglecting application execution security. ITU-T YD/T regulates data access security, transmission protocol security, and system execution security but is inadequate concerning application execution security. In this study, NIST SP800-163.164 and ITU-T YD/T 2407 were used as the standards for data access security and transmission protocol security, whereas OWASP's top 10 weaknesses, the OWASP Mobile Security Guide, and the CSA's MAST were applied to address the shortcomings in application and system execution security.

3.2 Risk Assessment Framework

This study used the ISO 31010:2009 risk management–risk assessment techniques; their risk assessment procedures are as follows: identify risks and their causes, determine the consequences of risks, redefine the probability of risks, and identify the factors that reduce the consequences or likelihood of risks. High-level risk assessment practices were adopted according to the risk assessment methods and risk levels of information security testing for domestic mobile phone system built-in software [7]. As shown in Tables 1 and 2, Mobile Top 10 2016-Top 10 and Cloud Security Alliance were used to analyze risk assessment threat and evaluate privacy security, native security, and protection security in mobile devices as well as to evaluate the possible CIA impact levels for the risk assessment framework, for which the risk levels were divided into low, medium, and high.

Table 1. Risk assessment framework

Mobile Top 10 2016-Top 10										Category Distinction			Risk Level		
M1 Improper Platform Usage	M2 Insecure Data Access	M3 Insecure Communication	M4 Insecure Authentication	M5 Insufficient Encryption	M6 Insecure Authorization	M7 Client Code Quality	M8 Code Tampering	M9 Reverse Engineering	M10 Redundant Functions	Privacy Security	Native Security	Protection Security	Low	Medium	High

Table 2. Risk level

Risk level	Description
Low	The mobile device provides users with the basic relevant mobile device security and personal information protection, such as thorough information on mobile device resources and authorized usage and the provision of relevant protections
Medium	The mobile device provides a complete protection mechanism for user information and privacy, such as security mechanisms for the data transmission process and data storage, ensuring users' security protection throughout use, transmission, and storage
High	The mobile device should ensure that the core layer is not maliciously tampered with and compromised to avoid information leakage

4 Results and Discussion

4.1 Test Item Details

This study addressed the inadequacies of mobile application specifications by considering mobile application items regulated in various countries, and the details of the test items are described in Table 3.

The six test items for the data leakage dimension included how user consent or declination is used and acquired when an application launches, rejection mechanism, and how applications access unrelated services without user authorization.

Additionally, test items on privacy and information security included whether sensitive information is stored in the file content of mobile applications and whether a password strength policy is applied to avoid information theft; in total, six test items were specified for this dimension.

The test items for the API/library native dimension included whether mobile applications avoid incorporating reverse engineering and security vulnerabilities during application development, which can cause native security threats; seven test items were specified for this dimension.

Table 3. Test item details for inspection specifications for mobile applications

Test item	Security requirements	
A.1 Inappropriate Authority Extraction	The mobile application should fully declare the instructions for mobile device resources and authorized uses and obtain the relevant authorization from the users	
Test items	Test number	Test details
A.1 Inappropriate Authority Extraction	A.1.1	A complete description of access to sensitive data, mobile device resources, and declaration of authorized uses should be provided during the launch of the mobile application
	A.1.2	Access to sensitive data related to personal information by the mobile device should be used to determine whether the application provides the relevant identity authorization mechanism
	A.1.3	The mobile application should acquire user consent before accessing sensitive data
	A.1.4	The mobile application is permitted to access functions without user authorization
	A.1.5	Once the mobile application is denied access to sensitive data by users, the application may not access sensitive data through other means
	A.1.6	Users' use of mobile application functions such as contacts or message sending, receiving, and deleting should be recorded by the server. Testers should try to access another user function to verify whether they can access functions that should not be permitted by the users' role/privilege (but may be permitted for other user account types)

The test items for the application data security dimension included whether the data required for generating information after installing the application undergoes a memory dump and whether the application automatically closes or locks within a configured period of inactivity to mitigate data security problems linked with specific conditions or loopholes designed during development; five test items were based on application data security.

Four test items were formulated for the native environment obfuscation dimension and were based on the security risks caused by unexpected conditions accidentally produced by applications and the repackaging and obfuscation techniques.

Fourteen test items were formulated for the transmission protocol and encryption strength dimension, including whether applications completely encrypt data for the transmission protocol or channel during transmission, adopt secure encryption algorithms and authentication mechanisms to prevent data theft during transmission.

Four test items were specified for the data storage security dimension, including whether applications provide encryption when they store sensitive data and whether passwords are saved in a protected area of the operating system and in encryption form

Table 4. Procedure inspection and methods for improvement for mobile application specifications

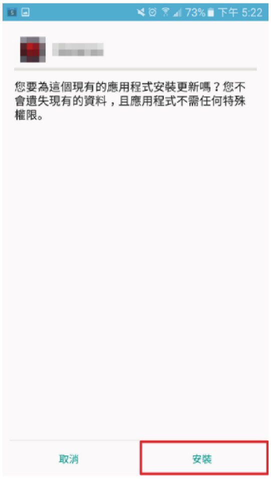
Test Number: A.1.4

Test Procedures:

Check the user instructions in the application privacy section of the mobile device application to ensure that it corresponds to the resource usage instruction specified in test item A.1.1 that authorizes users on the authorization list. Offering an authorization means that the application is authorized to use the resources when its authorized status is set as default or until the application is removed, although the application should be inspected to determine whether it uses functions without user authorization.

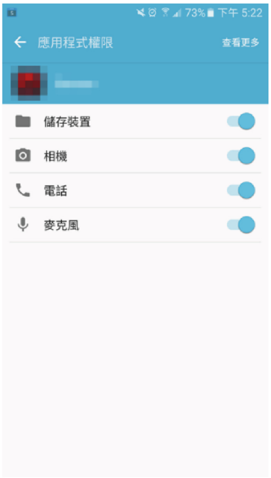
Example Screenshots:

The image below shows the usage statement for the privacy section:



您要為這個現有的應用程式安裝更新嗎？您不會遺失現有的資料，且應用程式不需任何特殊權限。

The image below shows the default authorization when user authorization has not been granted:



儲存裝置 ●
相機 ●
電話 ●
麥克風 ●

Methods for Improvement:

The tester inspects the declared authorization of the <uses-permission> used in the AndroidManifest.xml file using the disassembler to determine whether use of functions that are not authorized by the user occurs.

to avoid sensitive data from being accessible as plain text in executable files and prevent sensitive data from being acquired without authorization.

4.2 Procedure Inspection and Methods for Improvement

Table 4 shows the improved mobile application specifications proposed in this study. The improvement recommendations were proposed in accordance with the test procedures described in the test dimensions and the methods of improvements suggested in the mobile application specifications of each country. They comprised improper authority extraction, private data security, API/library native, application data security, native environment obfuscation, transmission protocol and encryption strength, and data storage security, and encompassed 46 test items. The test procedures, example screenshots, and methods for improvement are listed on the basis of the test details. The actual testing was conducted by users and developers according to the test procedures of each test item. Images were simultaneously used as examples to enable users and developers to understand the purposes of the test items, and methods for improvement were provided to reduce the risks of mobile device security.

4.3 Risk Assessment Matrix

According to Table 5, Mobile Top 10 2016-Top 10 and CSA were used for risk assessment and threat analysis when privacy security, native security, and protection security of the mobile device were evaluated to assess the possible CIA impact level for the risk assessment framework. Three risk levels were used: low, medium, and high.

Table 5. Risk Assessment of Mobile Devices

OWASP mobile TOP 10	Test number	Test details	Risk levels
M1 Improper Platform Usage	A.1.1	A complete description of access to sensitive data, mobile device resources, and declaration of authorized uses should be provided during the launch of the mobile application	Low
M4 Insecure Authentication	A.1.2	Access to sensitive data related to personal information by the mobile device should be used to determine whether the application provides the relevant identity authorization mechanism	Low
M6 Insecure Authorization	A.1.3	The mobile application should acquire user consent before accessing sensitive data	Low
M6 Insecure Authorization	A.1.4	The mobile application uses functions without acquiring user authorization	Medium

5 Conclusion

The popularity of mobile devices has resulted in increasing dependence on them, and mobile application developers' lack of security concern risks users' personal information leakage; thus, the mobile device specifications in Taiwan remain inadequate. The literature on mobile device testing specifications were reviewed in this study and were classified into specification priorities for data authorization security, transmission protocol security, application execution security, and system execution security. The OWASP, NIST and CSA white paper on mobile device specifications were used as the specification standard. The test specification dimensions for improving mobile applications were improper authority extraction, privacy data security, API/library native, application data, native environment obfuscation, transmission protocol and encryption strength, data storage security, for a total of 46 test items. Recommendations were proposed in accordance with the test procedures in the test dimensions, methods for improvement, and risk assessment so that domestic mobile device specifications can be improved.

The research content provided improvements for the domestic mobile device specifications, and studies on other countries and research and development technology can be used in the future for analysis, to render the testing specifications for mobile devices more effective and capable of protecting user security.

References

1. OWASP Top 10 Proactive Controls (2016), https://www.owasp.org/images/5/57/OWASP_Proactive_Controls_2.pdf
2. Introduction to NIST, https://www.nist.gov/about-nist/our-organization
3. Introduction to the Industrial Development Bureau, Ministry of Economic Affairs, https://www.moeaidb.gov.tw/external/view/rwd_tw/intro/index01.html
4. Introduction to the National Communications Commission, http://www.ncc.gov.tw/chinese/content.aspx?site_content_sn=2255&is_history=0
5. Introduction to Cloud Security Alliance, http://www.twcsa.org
6. Smith, M.: Computer Security-Threats, Vulnerabilities and Countermeasures, Information Age, pp. 205–210, October 1989
7. National Communications: 2015 Report on Information Security Inspection of Built-in Software for Mobile Phone Systems 104 (2016)

Application of Pattern for New CAPTCHA Generation Idea

Thawatwong Lawan(✉)

Department of Computer Science, Faculty of Informatics, Mahasarakham University,
Kamrieng, Kantarawichai, Mahasarakham Province, Thailand
thawatwong@gmail.com

Abstract. Application of pattern for new CAPTCHA generation idea aims to present concept that applies mathematics theory. In this study, pattern is chosen for CAPTCHAs generating. There are 400 participants who approaching this study on the internet. Three type of pattern CAPTCHAs with two sample were study. There are shape pattern, color pattern and shape-color pattern. Amount of first correct answers, amount of total answers, percentage of success, amount of spent time and five point usability score were collected. The result shows that the most amount of first correct answers is color-shape pattern at 363. It also conforms to Color-shape CAPTCHA which shows the highest percentage of success at 97.06. In the amount of spent time, shape-color pattern CAPTCHAs indicates least time to solve at 5.25 s. The total spent time to find the correct answer of all type CAPTCHAs are 5.25 to 8.87 s. Usability score result shows that shape pattern CAPTCHAs is the highest score in all aspects at 4.34 with non-difference significant at p-value < 0.01. The approach rates over level 4.00 in all, which means the approach feels all type of Pattern CAPTCHAs practical is useful.

Keywords: Pattern · Geometric shape · CAPTCHAs

1 Introduction

Nowadays, there are various internet-enabled online services being used in our daily life such as internet banking, e-commerce, online subscription and data uploading or downloading. These resources can be easily accessed by the malicious users. These users have developed automated or semi-automated software (bots) that can mimic (simulate) human operation in order to access to those services. Most of these kind of intrusions aim to agitate, attack or destroy data in the network system. As the result of that, a lot of problems can happen for example, it can flood the online form, over subscribe the amount of members in the website, create abusive accounts, and join the online surveys and so on. These problems not only skew the results but also damage the system. There are a lot of effort to overcome the problems of bots simulating human operation. The well-known approach is using the CAPTCHA to distinguish actions between automated bots and human. CAPTCHAs, the automated attack protection mechanism, were presented by von Ahn et al. in 2003 [1]. It was designed to distinguish human operation from bots by using simple questions that cannot be quickly solved by bots but easily for

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 257–264, 2018.
https://doi.org/10.1007/978-3-319-76451-1_24

human. There are many kind of CAPTCHAs such as text-based, image-based, audio-based, and multimedia-based CAPTCHA [2, 3]. CAPTCHAs has been effective method that can use for preventing various online services from automated or semi-automated software (bots) until now. Thus, in last a few years, CAPTCHAs were not the first choice of method that every web service chose for prevent their web from bots. That is because there are many bots type that can attack CAPTCHA such as the character recognition. It is an automated computer image processing technique which is used extensively by spammers. It can defeat text-based CAPTCHAs and the image-based CAPTCHAs for example, 3D CAPTCHA and puzzle CAPTCHA. These are the advancement of algorithm and image processing technology today that lead automated bots to overcome these challenges easily. Therefore, CAPTCHAs developer tries to build all CAPTCHAs more complicated and create system load such as text-based CAPTCHAs, which there are more warping and distraction of the letters, and Image-based CAPTCHAs, which there are more images stored in big databases.

As a result, this situation makes sure that the generation state can consume more time to access and use more storing spaces of disk [4–7]. For these reasons, CAPTCHs become the strong guard that does not allow both human and automated bots to access web services. The complicated CAPTCHAs are not correct for solving bots problem. In this paper, the researcher tries to present new generate CAPTCHAs idea from very simple pattern in mathematics in order to develop CAPTCHAs.

2 Overview of Our Scheme

2.1 Pattern in Mathematics

The definition of pattern in mathematics is regularly repeated in arrangement of shapes, colors, or lines on a surface [8]. Mathematician sometimes calls mathematics the Science of Pattern. Pattern is very simple and there are many patterns all around. Everybody knows about Patterns and used to solve them such as color pattern, shape pattern and number pattern [9, 10]. The excellent method to solve pattern is to observe sequence, find hidden rules, then find the correct answer. Solving pattern does not require difficult mathematic theories or any language. The correct answer shows in the line of sample in pattern sequence. Therefore, finding and understanding patterns can provide more experiences in solving them. Patterns can help us learn to predict the future, discover new things and understand this world better [9–11]. From all characteristics of pattern that consist of good characteristics of CAPTCHAs should be: (1) easy for most people to solve (2) difficult for automated software to solve and (3) easy to generate and evaluate [11]. Therefore, in this study, pattern is used to generate CAPTCHAs. In This case, the researcher chooses three types of pattern which are shape pattern, color pattern and combined shape-color pattern to generate CAPTCHAs.

2.2 CAPTCHAs Generated

The technique of randomly the pattern creation uses colors and geometric shapes to generate the pattern by sequencing into series. After that, it will be set as the reference pattern model. The authenticity process is to pass the test so users must find the missing answer by learning from the sequence of reference pattern model. This mechanism is easily manageable for human to find out, which the correct answer should be the next in sequence of series. The CAPTCHAs are generated by simple algorithm programming. The images do not need to be stored in the databases because color and geometric shape are already existed. Hence, it takes very less time in the process of CAPTCHAs generation. This function can be attached to main program during implementation state. The flow chart of challenge generating is shown as Fig. 1.

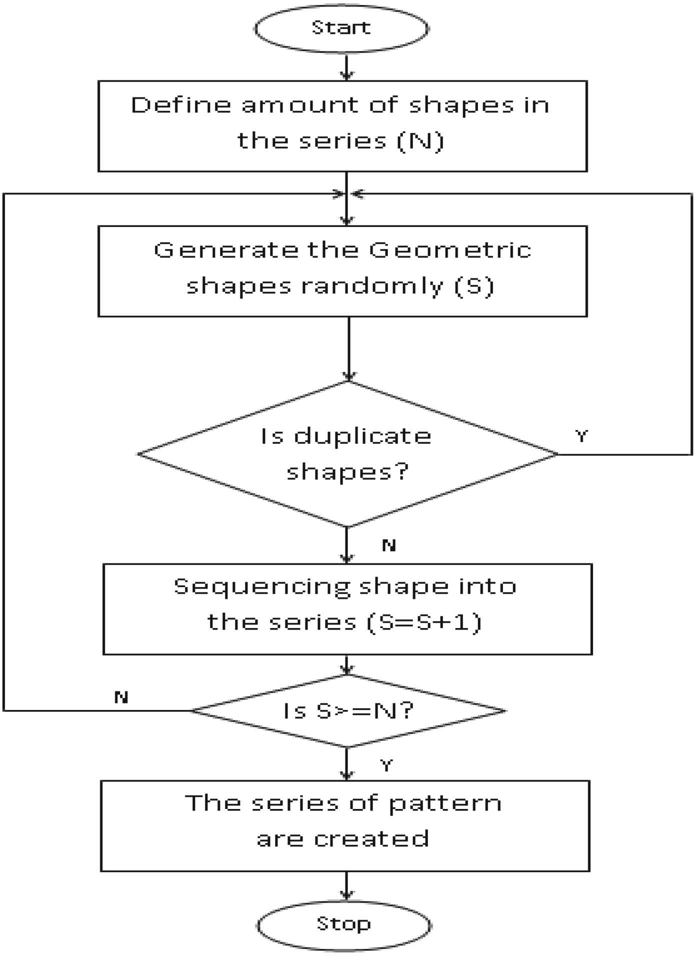

Start
Define amount of shapes in the series (N)
Generate the Geometric shapes randomly (S)
Is duplicate shapes?
Y
N
Sequencing shape into the series (S=S+1)
N
Is S>=N?
Y
The series of pattern are created
Stop

Fig. 1. Flowchart of challenge generation.

2.3 Methodology and Implementation

In this study, the proposed scheme is deployed in the main program of the website for everyone who would like to participate. An experimental of the proposed scheme is designed by renting website hosting services as the web-server. As for the development of Pattern CAPTCHA, JAVA script and PHP are used on internet platform. There are three types of pattern CAPTCHs in this study: (1) shape pattern (2) color pattern and (3) shape-color pattern. There are 2 sample sequences in all types of pattern CAPTCHAs in this study. Then, the study also conducts a survey using questionnaires, which are provided to access online on the website. The connection schema is illustrated as Fig. 2. There are three parts in CAPTCHAs webpage. Part one is general data of the approaches about sex, age, education level were collect. Part two is pattern CAPTCHAs, which in this part, CAPTCHA data about amount of time and a number of answers are collected. The approach will be analyzed and find the answer of CAPTCHA. Then, they will evaluate themselves. There is no language to explain the relationship of pattern series. And part 3 is evaluation part, which the approaches will rate usability score. This study uses 5 point scale to represent usability of pattern CAPTCHAs. The three types of pattern CAPTCHAs are shown in Fig. 3.

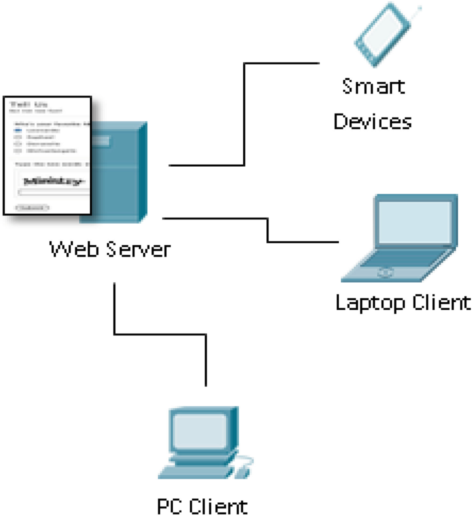
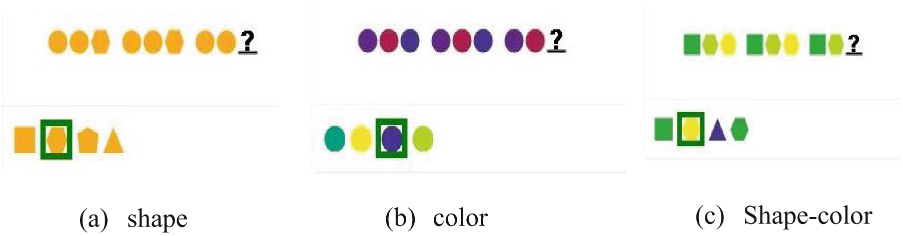


Fig. 2. Implementation layout.



(a) shape (b) color (c) Shape-color

Fig. 3. Three type of pattern CAPTCHAs

The approach will observe and try to comprehend the relation of pattern series. Then, it will choose the correct answer. There is no limit of time to find answers for all of the questions. After that, data about the amount of time to answer the questions, number of all answers, number of first correct answers, numbers of wrong answers, and satisfied score for Statistical Analysis will be collected. The three types of pattern with two sample sequences CAPTCHAs are shown in Fig. 3.

Amount of the approaches refers to the Electronic Transactions Development Agency (Public Organization), or ETDA survey. The result of a number of Thailand Internet User Profile 2016 were 38,015,725. Then, the sample's size is calculated by TARO YAMANE formula [12].

$$n = N/1 + N(e)^2 \tag{1}$$

where,

n = Sample size
N = Population size
e = except error size = 5%

$$\begin{aligned} \text{So Sample size} &= 38,015,725/1 + 38,015,725(0.05)^2 \\ &= 400 \end{aligned}$$

Data analysis in this study is amount, percentage, means and standard deviation and statistic. For analysis, SPSS for Windows Version 10.0, a program for statistic analyze is used.

3 Result and Discussion

The result of the study of Application of Pattern for new CAPTCHA generation idea is shown in 3 tables:

From the Table 1, the result shows the amount of first correct answers, amount of total answers and percentage of success in different types of pattern CAPTCHAs. The result shows that the approach can solve color-shape pattern better than other types of pattern CAPTCHAs. Color-shape pattern has the most amount of first correct answers at 363. It can conform to Color-shape CAPTCHA, which shows the highest percentage of success at 97.06. However, all types of pattern CAPTCHAs have percentage of success over 90. This result supports that pattern can apply to generate CAPTCHAs. Moreover, the approach can understand and show correct response to pattern CAPTCHAs immediately. In this study, no language is used for explaining how to pass the CAPTCHAs. Therefore, this can be the strength of pattern CAPTCHAs which suggests that they can communicate without any language. Furthermore, the remarkable of this result is the approach can solve pattern that have more than one factor. From this study, only shape or color pattern show lower percentage of success than color-shape pattern CAPTCHA. It can be assumed that more factors may provide more clues to the approach.

Table 1. Number of first correct, number of total answer and percentage of success in differences type of pattern CAPTCHA's.

CAPTCHA type	No. of first correct	No. of total answer	% of success
Shape pattern	358	381	93.96
Color pattern	357	389	91.77
Color-shape pattern	363	374	97.06

The result shown in Table 2 suggests that there is significant of different amount of spent time to solve pattern CAPTCHAs ($p < 0.05$). The approach spends less time to solve CAPTCHAs in shape-color pattern at 5.25 s ($p < 0.05$). However, in other CAPTCAHs types, the approach spends less than 10 s to solve CAPTCHAs.The total amount of spent time on finding the correct answer are 5.25 to 8.87 s. And there are a lot of research showing the spent time that users lose for solving CAPTCHAs such as Elie Bursztein and et al. in 2010, which suggests that average amount of time to solve the questions are 10.13 to 16.30 s [13]. Similarly,Youthasoonthorn Passzarkorn in 2014, which aims to suggest the evaluation of CAPTCHAs efficiency in www.captchachallenge.com. It indicates that means of spent time of the multination approach to pass CAPTCHAs are about 10.13 to 19.04 s [14]. Therefore, this may be the pattern CAPTCHAs which is suitable for CAPTCHAs developers. In addition, this result suggests that more factors will help the approach gets more hints for understand the rules of pattern CAPTCHAs. Furthermore, the result also shows the same trend to percentage of success. That is more factors will make the approach spends less time. After all, this result supports that in pattern queries about the attributes of things, the approach will apply reasons to the answer "What's next?" and human will develop "function sense" without simply asking for the next shape in the pattern.

Table 2. $\bar{x} \pm SD$ of time to find correct answer in different types of pattern CAPTCHAs.

CAPTCHAs type	$\bar{x} \pm SD$ (sec)
Shape pattern	8.87 ± 7.91^a
Color pattern	6.95 ± 19.99^b
Color-shape pattern	5.25 ± 5.13^b

[a] *Means values are express ± standard deviation.*

[b] *Difference letter at same line indicated statistic difference according Duncan test ($p < 0.05$).*

The usability score is shown in Table 3. There are four aspects that are evaluated by the approach of this study. All aspects that selected for used in this study are important because they are good characteristics of CAPTCHAs. Feeling of the approach can present users' senses which sometimes can represent reality more than machines can. From the table, it shows that the approach rates over 4.00 in all. That means the approach feels that all types of Pattern CAPTCHAs practical are useful. Shape pattern CAPTCHAs shows the highest score in all aspects which is 4.34. However, there is non significant in score differences when the statistic test is used. This result shows that pattern can apply for CAPTCHAs generating. It also suggests that everybody may get used to pattern in mathematic when they were young, or it prove that there are patterns all around us. Therefore, pattern can communicate itself. It can make the approach feels good when they see it in CAPTCHAS and solve it without any obstacle and getting annoyed.

Table 3. $\bar{x} \pm SD$ of usability score in difference type of pattern CAPTCHAs.

Usability aspect	Shape pattern	Color pattern	Shape-color pattern	Significant
	$\bar{x} \pm SD$	$\bar{x} \pm SD$	$\bar{x} \pm SD$	
Simplicity	4.40 ± 0.87	4.320 ± .90	4 ± 3610.87	0.455
Understandability	4.0 ± 28.87	4.0 ± 21.89	4.0 ± 22.85	0.514
CAPTCHAs size	4.240 ± .85	4.0 ± 22.84	4.0 ± 23.86	0.941
Solve time usage	4 ± 12091	4.0 ± 10.91	4.0 ± 09.93	0.915
All aspect score	4 ± 340.77	4.0 ± 27.87	4.0 ± 29.87	0.508

Means values are express ± standard deviation.
Same letter at same line indicated statistic difference according Duncan test ($p < 0.05$).
Meaning of five point scale: 0.00–1.00 = very poor, 1.01–2.00 = poor, 2.01–3.00 = fair, 3.01–4.00 = good, 4.01–5.00 = excellent.

4 Conclusion

This research aims to apply pattern in mathematic for CAPTCHAs generating to find new way for develop CAPTCHAs that is easy to understand and friendly with human, and hard for bots in web service to use. This research suggests the good choice which pattern for generating CAPTCHAs is chosen by amount of first correct, percent of success, spend time to use and usability score. All results indicate that Pattern may be new choice to use for generating CAPTCHAs in the future because it shows that high amount of first correct answers and percentage of success are more than 90. Pattern CHAPTCHAs can decrease spent time for solving CAPTCHAs in comparison to other research. And the approach feels that pattern CAPTCHAs contains high usability. However, there are more issues in applying pattern for generating CAPTCHAs. In further study, the appropriated sample sequence of pattern, amount of factor to use and how many shape or color in pattern sequences of CAPTCHAs should be studied.

References

1. Von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: using hard AI problems for security. In: Proceeding of Eurocrypt, pp. 294–311. Springer (2003)
2. Singh, V., Pal, P.: Survey of different types of CAPTCHA. Int. J. Comput. Sci. Inf. Technol. **5**(2), 2242–2245 (2014)
3. Abdullah Hasan, W.K.: A survey of current research on CAPTCHA. Int. J. Comput. Sci. Eng. Surv. **7**(3), 1–21 (2016)
4. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: 14th International Proceedings of ACM CCS 2007, pp. 366–374. ACM, New York (2007)
5. Yan J., Ahmad, A.S.E.: Usability of CAPTCHAs or usability issues in CAPTCHA design. In: Symposium on Usable Privacy and Security (SOUPS 2008), pp. 44–52, Pittsburgh, PA, USA (2008)
6. Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: RECAPTCHA: human-based character recognition via web security measures. Science **321**(1), 1465–1468 (2008)
7. Gossweiler, R., Kamvar, M., Baluja, S.: What's up CAPTCHA?: A CAPTCHA based on image orientation. In: 18th International Conference on World Wide Web (WWW 2009), pp. 841–850. ACM, Madrid, Spain (2009)
8. Cambridge Advanced Learner's Dictionary & Thesaurus. http://dictionary.cabridge.org/dictionary/english/patternDefinitionof"pattern"/. Accessed 09 Mar 2017
9. http://www.mathsisfun.com/algebra/patterns.html. Accessed 09 Mar 2017
10. The Annenberg Foundation. https://www.learner.org/teacherslab/math/patterns/word.html. Accessed 09 Mar 2017
11. Komatsu, H., Ideura, Y.: Relationships between color, shape, and pattern selectivity of neurons in the inferior temporal cortex of the monkey. J. Neurophysiol. **70**(2), 677–694 (1993)
12. Thailand Internet User Survey 2016. Electronic Transactions Development Agency (Public Organization). https://www.quora.com/What-is-Yamane-sample-calculation. Accessed 09 Mar 2017
13. Bursztein, E., Bethard S., Fabry, C., Mitchell, J.C., Jurafsky, D.: How good are humans at solving CAPTCHAs? A large scale evaluation. homepage. https://web.stanford.edu/~jurafsky/burszstein_2010_captcha.pdf. Accessed 09 Mar 2017
14. Youthasoontorn, P., Phaibulpanich, A., Piromsopa, K.: Evaluation of CAPTCHAs efficiency. J. Inf. Technol. Appl. Manag. **22**(3), 55–64 (2015)

Digital Forensics and Mobile Systems

An Automatic Approach of Building Threat Patterns in Android

Chia-Mei Chen[1](✉), Yu-Hsuan Tsai[1], and Gu-Hsin Lai[2]

[1] National Sun Yat-Sen University, Kaohsiung 804, Taiwan
cchen@mail.nsysu.edu.tw
[2] Taiwan Police College, Taipei 116, Taiwan

Abstract. Nowadays, handheld devices have become popular but volume of malwares on mobile platform has also grown rapidly. To detect mobile malware, static approaches and dynamic approaches are two common ways used to analyze suspicious applications. Dynamic approaches detect malware base on the actual behaviors of applications, but how to trigger malicious behavior and the efficient of dynamic approaches are the difficulties of this kind of approaches. Due to the limited resource of mobile devices, static analysis approach is the practicable way to detect malwares on mobile device. Anti-virus software is the typical paradigm of static analysis approach. However, the effectiveness of Anti-virus software rely on its signatures. How to find an efficient and automatic way to build thread pattern of mobile malware is a critical issue to detect new or zero-day malware.

In this paper, a detect mechanism based on data flow is proposed. The proposed system analyzes the function calls and the data flow to identify malicious behaviors in Android mobile devices. Machine learning approach is used to build threat patterns automatically within a great volume of applications. The experimental result shows that the proposed system could detect malware with high accuracy and low false positive rate.

Keywords: Android malware · Data flow · Machine learning

1 Introduction

Recently, mobile devices like smartphone or tablet have become popular and powerful, more service or applications have been developed. People now install e-banking, e-shopping or social network apps on their own mobile devices, and some valuable or confidential data are also stored in their device. Therefore, mobile device have become new target for attackers for the purpose of financial gain. According to Tread Micro's investigation [12], 17 malwares had already being downloaded about 700,000 times before they had been removed. For example, FakeInstaller, a widespread mobile malware family, sends SMS messages to premium rate numbers without the user's consent [7]. Faketoken [10], a new malware, it could record phone calls, intercept text messages and steal data from various apps, including banking apps.

Among mobile platform (Android, iOS, windows mobile or Symbian), Android platform occupies the largest market share. According to IDC's latest study, Android

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 267–274, 2018.
https://doi.org/10.1007/978-3-319-76451-1_25

platform occupies about 85% market share in 2017 Q1 [6]. Therefore, more and more Android malware are created for the purpose of financial gain.

To detect malware, threat patterns are used to identify is a suspicious app is malicious or not. There are two main ways used to build threat patterns from collected malwares, they are dynamic analysis and static analysis approach. Dynamic analysis first executes malware in a controlled environment (in most case, in a virtual machine), then, analysts observe and record malware's behavior for building threat patterns. However, authors of malware may use anti-VM technique to evade detecting such as sleep for a while before actually doing some malicious behaviors. Besides, to observe and record malware's behavior, dynamic analysis systems need many resources (Memory space, disk space and computational resources). Either in a powerful workstation or in a smartphone does not have enough resource to run dynamic analysis for the great volume of applications. Rather than executing malware, static analysis can analyzes malware without executing it. One of the advantage of static analysis is that it can scan and check malware quickly. Due to this reason, we use static analysis approach to detect malware in Android system.

Anti-virus software is most common used static analysis approach. By means of matching signatures, anti-virus could detect malware efficiently. However, creators of malware develop variants to evade detection by anti-virus. In this paper, an automatic approach to build threat pattern on Android platform is proposed. The proposed system use reverse engineering and machine learning approach to build threat pattern automatically.

To detect variants, the threat patterns in the proposed system are based on the concept of data flow instead of specific strings. The structure of the paper is as follows: Sect. 2 reviews the literature of mobile malware detection approaches, Sect. 3 presents the details of proposed system, Sect. 4 discusses the experimental results, and Sect. 5 concludes the paper.

2 Related Work

Sarma et al. [8] proposed an approach that analyzing risk of application based on permission. Cerbo et al. [3] using Apriori algorithm to analysis the permission's subset of applications, which are same type. If an application request a permission, which is different from the subset, it might be malware. It not good to detect malwares only based on analyzing permissions because there are some drawbacks [1]. Most app developers produce over-privileged mobile software [8, 14], permission based approach might not be enough to identify mobile malware. Moreover, Grace et al. [5] discovered that malware could perform malicious behaviors without asking for the permissions.

Enck et al. built an Android sandbox by modifying Android's source code. The sandbox monitor the leakage of sensitive data in device. For example, if a sensitive data such as IMEI or DeviceId appear in text message or Internet, the system will issue an alert. The system could detect possible information leakage; however, the proposed system could not identify which application is suspicious. Shabtai et al. [9] proposed a detection system applying knowledge-based and temporal abstraction method to identify unknown malware. Shabtai's system built temporal threat patterns by logging history events (for example, SMS message or the installation of apps). In Shabtai's work, interaction between users and devices is an import criteria for detecting malware. For example, sending a SMS message without interaction with user is identified as abnormal behavior. However, malware can easily avoid this approach by using social engineering techniques. Moreover, monitoring every events in user devices needs a lot of resource. There are some challenges for dynamic based detection approaches in Android platform. First, dynamic analysis needs a lot of time and resources, but there are millions applications in official market and third-party markets, so it is impractical to scan all applications. Second, whether system can trigger malicious behaviors is also a problem because the analysis process may terminate before the malicious behaviors occur. These challenges all make dynamic analysis hard to detect malware effectively.

Wu et al. proposed an approach [10] that gets permission and component information from Manifest file, and then extracts information of Intent, API calls and communication between components from source code, then using k-means algorithm and expectation–maximization algorithm to classified applications. Yerima et al. proposed an approach [15] that first using feature selection techniques to find out API calls and system calls that are proper to distinguish malwares and benign applications, then using Bayesian classifier to classify malwares and benign applications. It seems that API or system calls are important to identify malware. To detect new variants, Chen proposed an approach which use concept of data flow to build threat patterns for mobile malware. However, the threat patterns in Chen's work are built manually [4]. Building a threat pattern manually is a time consuming task. Therefore, how to build threat patterns automatically is critical for detecting mobile malware.

3 Proposed Approach

3.1 Threat Patterns Built by the Concept of Data Flow

In this paper, we proposed an automatic approach to build threat patterns for detecting mobile malware. To detect variants, the concept of data flow is used for building threat patterns. The threat patterns consist of two main components; they are sensitive data and sensitive methods. One application is identified as malicious when sensitive data flow into sensitive methods. Sensitive data may be an API or constant variables in source code. However, building such attack patterns is a time consuming task. Thus an automatic mechanism for threat pattern building is needed. An automatic approach to build threat patterns

In this paper, we propose an automatic mechanism to build threat patterns for Android malware. Figure 1 illustrates the process of proposed mechanism.

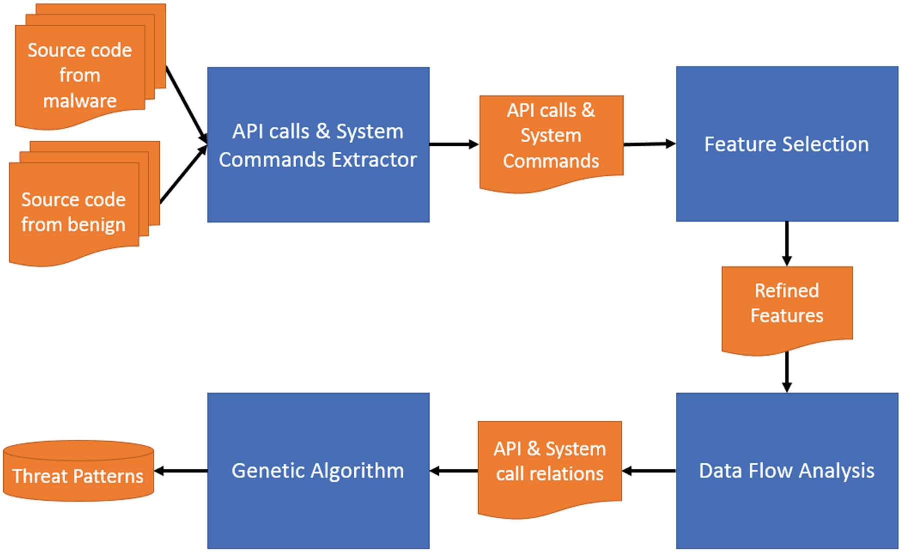


Fig. 1. Threat pattern building process.

The first input of the proposed system is source code of either malware or benign. In this paper, reverse engineering technology is used to get source code from APK files. Figure 2 illustrates the process of getting source codes from APK files. In this paper, we use three tools to decompile Android APK files. They are APKTool, dex2jar and JAD. APKTool which can help us getting .dex files from apk file, dex2jar which can transform .dex files from APKTool into .class files, and JAD which is a decompiler which can transform.class files into .jad files. These .jad files are the Java source code of APK files.

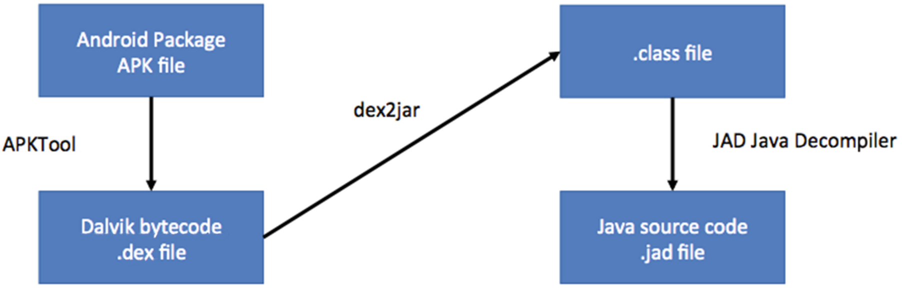


Fig. 2. Reverse engineering process

The first component of proposed system is API calls and system commands extractor, which can get API calls and system commands from source codes. The extracted API calls and system commands will be used to train sensitive API calls, system commands and data which malicious application tend to use. There are 21,193 API calls and 258 system commands from official Android web site. Most API calls are useless for analyzing malware behaviors and the great volume of API calls will also slow down the speed of analysis. In this paper, mutual information is used to get sensitive API calls, system command and data. There are 19 API calls and system command are choose as sensitive API calls to build threat patterns.

After sensitive API and system commands are defined, the proposed system will perform data flow analysis. The proposed system try to find all data flow from collected APK files. Each data flow contains a ID, source data and at least one sensitive API call or system command. Table 1 illustrates an example of data flow.

Table 1. An example of data flow

Data flow ID	Sensitive data	Sensitive API calls or system command
1	getSubScribeID	sendTextMessage
2	getLine1Number	sendTextMessage

There are 3,487 data flows are found in the proposed system; it means that there are 3,487 possible threat patterns to identify mobile malware. The following step is to select useful threat patterns from these data flows. Checking the effectiveness manually of each data flow is a time consuming task. In this paper, genetic algorithm is used to find a set of threat patterns from these data flows. To implement genetic algorithm, we need to encode our solution into a gene which is usually a binary string. We give index to every data flow, if the data flow has an index k, and this relationship has been choose, the value of the binary string at index k will be 1, otherwise will be 0. The length of the binary string will be as same as the number of data flow. For example, if the binary string is 000110 means there are total six data flow in proposed system, and the 4th and 5th are being selected as threat patterns.

There are some important parameters in genetic algorithm, they are population, mutation probability, number of iteration and fitness function: There parameters will influence the solution and the system performance. After carefully test, the parameters are illustrates in Table 2. Fourteen threat patterns are selected in this step.

Table 2. Parameters of genetic algorithm in proposed system

Parameter name	Values
Population	100
Mutation probability	0.1
Number of iteration	40
Fitness function	True positive + Precision

3.2 Detection of Malware by Data Flow Based Threat Pattern

Now, our threat pattern generation process already defines several threat patterns. The proposed system could use these threat patterns to detect Android malware. Figure 3 illustrates the detection process.

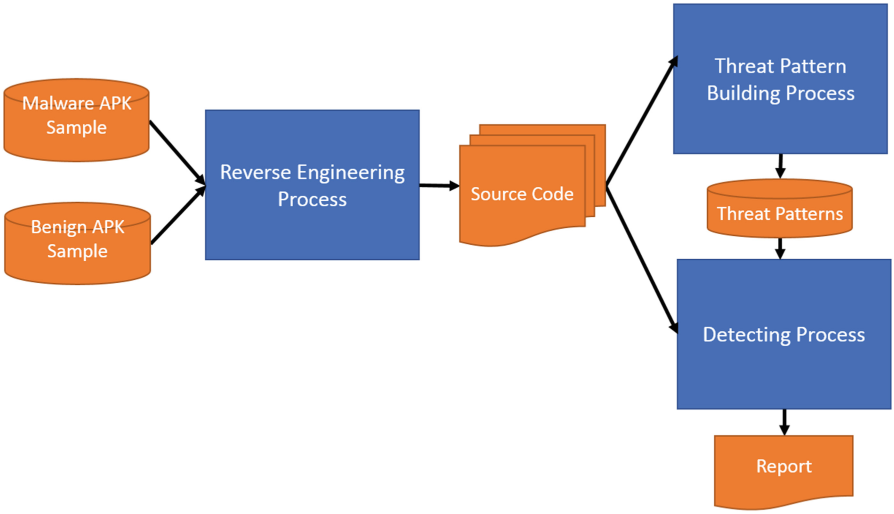


Fig. 3. Detection process of proposed system.

The first step of detection process is reverse engineering process, all APK files are decompiled into java source files. Then, the proposed find all data flows in test sample. Given A is a test sample, B is a set of threat patterns in the proposed system and C is a set of data flows of A. D$ataFlow(R_1), R_1 \in B$ D$ataFlow(R_2), R_2 \in C$ $\exists J(R_1, R_2) > Threshold \rightarrow$ A is malicious where similarity function is defined as Eq. 1.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

After out carefully test, threshold is set to 0.75 to get best performance in our system.

4 System Evaluation

Three experiments were performed to evaluate the detection performance of the proposed system. To evaluation performance of proposed, 1,259 malwares and 1,259 benign applications are used. The benign applications were downloaded from Google Play market, and our benign applications consist by the top popular free apps in each category. There still have a chance that Google Play market contain some malwares, so the benign applications we downloaded must had been existing on Google Play market for over three month, because we believe that the chance of an malicious and popular application can survive in Google Play market for over three month is low. The malwares in this evaluation comes from Android Malware Genome Project [2], 1,259 malwares in total from 49 different families. The analysis result show in Table 3, the result shown the proposed system could detect mobile malware well.

Table 3. Result of performance evaluation

True positive	False positive	Precision
96.5%	10.6%	90.0%

5 Conclusion and System Limitations

In this paper, we developed it based on static analysis approach, so we can deal with large amount of applications. We also provide reports that contain the relationships of API calls and system commands, so researchers can easily find what malicious behaviors the application might conduct. Our approach also can generate threat patterns automatically, so researchers don't have to trace malware source code line by line, it can greatly shorten the time to build threat patterns. The threat patterns in the proposed system use the concept of data flow. Our threat patterns have three advantages. First, unlike threat patterns built by specific strings, our threat patterns could detect variants. Second, analyzers could understand the behavior of malware after examine the threat patterns. Third, analyzers could check the detection rate pattern by pattern. Analyzers could delete obsolete or low detection rate patterns.

There are still some limits in static analysis. Some malware might execute commands receiving from command and control server. Approach based on tracing source codes cannot detect malware if the malicious parts does not inside source codes.

Although we can reverse Android APK file back to source code, some application use NDK (Native Development Kit) for some special purpose. Using NDK, Android programmer can use C to develop some functionality and communicate with Java via JNI (Java Native Interface). The C source code will be compiled into share object (.so file) which is very difficult to decompile back to source code. If malware authors write their malicious code in C, we cannot detect it malicious behaviors through analyzing its Java source code.

References

1. Aafer, A., Wenliang, D., Heng, Y.: DroidAPIMiner: Mining API-level features for robust malware detection in android. In: International Conference on Security and Privacy in Communication Systems, pp. 86–103 (2013)
2. Android Malware Genome Project. http://www.malgenomeproject.org/. Accessed 21 Oct 2017
3. Cerbo, F.D., Girardello, A., Michahelles, F., Voronkova, S.: Detection of malicious applications on android OS. In: International Workshop on Computational Forensics, pp. 138–149 (2011)
4. Chen, C.M., Lai, G.H., Lin, J.M.: Identifying threat patterns of android applications. In: 12th Asia Joint Conference on Information Security, pp. 69–74 (2017)
5. Grace, M., Zhou, Y., Wang, Z., Jiang, X.: Systematic detection of capability leaks in stock android smartphones. In: Proceedings of the 19th Network and Distributed System Security Symposium (2012)
6. IDC: IDC Quarterly Mobile Phone Tracker. https://www.idc.com/promo/smartphone-market-share/os. Accessed 21 Oct 2017

7. McAfee Lab: FakeInstaller' leads the attack on android phones (2012). https://blogs.mcafee.com/mcafee-labs/fakeinstaller-leads-the-attack-on-android-phones
8. Sarma, B.P., Li, N., Gates, C., Potharaju, R., Nita-Rotaru, C., Molloy, I.: Android permissions: a perspective combining risks and benefits. In: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies (2012)
9. Shabtai, A., Kanonov, U., Elovici, Y.: Intrusion detection for mobile devices using the knowledge-based, temporal abstraction method. J. Syst. Softw. **83**(8), 1524–1537 (2010)
10. TechRepublic, new faketoken android malware records calls, intercepts texts, and steals credit card info. https://www.techrepublic.com/article/new-faketoken-android-malware-records-calls-intercepts-texts-and-steals-credit-card-info/. Accessed 21 Oct 2017
11. TrendMicro: "Android Malware: How Worried Should You Be?"
12. http://blog.trendmicro.com/trendlabs-security-intelligence/android-malware-how-worried-should-you-be/
13. William, E., Ongtang, M., McDaniel, P.: On lightweight mobile phone application certification. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 235–245 (2009)
14. Wu, D.J., Mao, C.H., Wei, T.E, Lee, H.M., Wu, K.P.: DroidMat: android malware detection through manifest and api calls tracing. In: Proceedings of 7th Asia Joint Conference on Information Security (2012)
15. Yerima, S.Y., Sezer, S., lliams, G., Muttik, I.: A new android malware detection approach using Bayesian classification. In: The 27th IEEE International Conference on Advanced Information Networking and Applications (2013)

Identifying Temporal Patterns Using ADS in NTFS for Digital Forensics

Da-Yu Kao(✉) and Yuan-Pei Chan

Department of Information Management, Central Police University,
Taoyuan City 333, Taiwan, ROC
camel@mail.cpu.edu.tw

Abstract. The storage and handling of alternate data stream (ADS) in NTFS have posted significant challenges for law enforcement agencies (LEAs). ADS can hide data as any formats in additional $DATA attributes of digital file. The process of data content will update some metadata attributes of date-time stamp in files. This paper introduces ADS and reviews the literature pertaining to the forensic analysis of its data hiding. It describes some temporal patterns for evaluating if ADS are hidden in digital files or not. The analysis of file metadata assists in accurately correlating activities from date-time stamp evidence. The results demonstrate the effectiveness of temporal patterns for digital forensics across various types of file operations.

Keywords: Alternate data stream · Date-time stamp · Digital forensics
Temporal patterns · NTFS

1 Introduction

This growing dependence of digital technology has been a bonanza to computer criminals to carry out their missions. Microsoft Windows systems have become one of the primary targets for cybercriminals. In digital forensics, file date-time stamps are vital attributes as they can establish the temporal sequence of events and time spans that can lead to crime reconstruction for investigations and for court proceedings [5]. This paper describes how alternate data stream (ADS) influences the date-time stamp attributes of digital evidence in Windows. The purpose of this paper is to improve the potential ability to discover temporal patterns, which are normally hidden to the human analyst.

1.1 ADS

ADS was introduced to make Windows new technology file system (NTFS) compatible with HFS file system of Macintosh. It is both a feature and vulnerability of NTFS and becomes one of the possible ways for hiding malware [9]. NTFS file system, which manages the data and its metadata, is a popular file system in Windows Operating System. In NTFS, the main data stream is usually visible to the user. The NTFS file system reserves space for the master file table (MFT) to contain information about a file, including its size, time and date stamps, permissions, and data content. Disk space

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 275–285, 2018.
https://doi.org/10.1007/978-3-319-76451-1_26

that has been allocated for these entries will not be reallocated, and the size of the MFT does not decrease [2, 8]. An MFT file record may have more than one $DATA attribute, and the additional $DATA attribute is ADS [9]. ADS provides additional descriptions for folders or files and attaches data streams to an NTFS file or folder. Proper user input examination and analysis become essential to defend against this ADS attack. The storage and handling of ADS create significant challenges for law enforcement agencies (LEAs).

1.2 Data Hiding in ADS

A suspect can hide the data or files on the NTFS file system with ADS so that they are not accessible to anyone. ADS can be used to hide data in NTFS file system for the following reasons [1–3]:

- No size limits: ADS does not have any size limits and several streams can be linked to each file.
- First $DATA attribute Examination: Most of the system utilities only examine the first unnamed $DATA attribute.
- No-Show: ADS does not show up in directory listing and the file size of original file does not change.
- ADS can embed metadata in any files or folders without altering their original functionality or content.

The literature reviews of three key attributes and their temporal values in NTFS $AttrDef File are discussed in Sect. 2. Section 3 describes experiment environment, observations, and the follow-up findings. Our conclusions are given in Sect. 4.

2 Literature Review

Temporal analysis of digital files is a crucial process that carries significant value to establish its sequence of file operations in the computer system [4]. Different file systems employ different types of date-time stamp mechanisms [5]. Even though different types of temporal analysis were proposed in many researches, there is still no ADS temporal analysis on the $SI/$FN attributes of NTFS. This paper tries to provide behavioral characteristics of date-time stamps in NTFS so that the temporal analysis in crime reconstruction can be explored to support or refute the chronological order of events. Moreover, the methodologies adopted for NTFS in this paper can also be applied in other file systems.

2.1 Three Key Attributes in NTFS

Digital files on NTFS are actually collections of multiple information, which contains volume attributes, certain properties, and their corresponding details. The NTFS $AttrDef file has three key attributes (Fig. 1): $STANDARD_INFORMATION ($SI for short), $FILE_NAME ($FN for short), and $DATA. Windows updates these attributes in diverse ways, which cause $FN date-time stamps are not consistent with $SI [2, 5, 6].

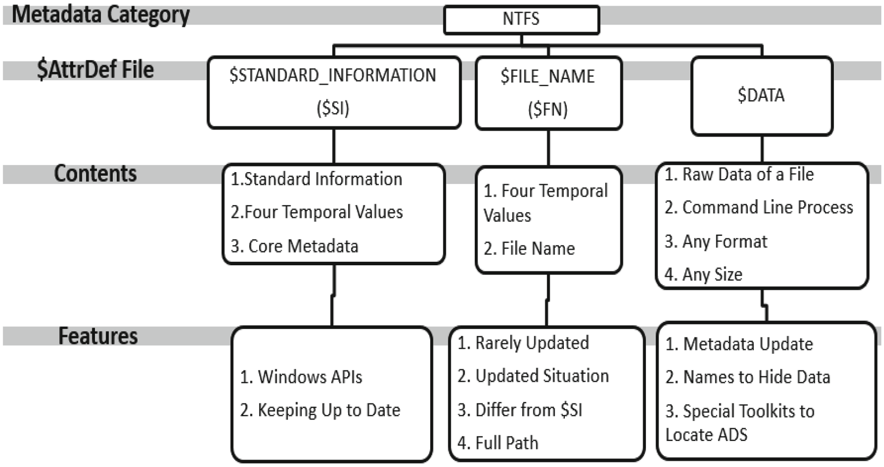


Fig. 1. Comparison among $SI, $FN, and $DATA

2.1.1 $SI Attribute: Keeping Up to Date

(1) Contents

- Standard information: Standard information about a file includes primary date-time stamps, ownership, security, and quota information.
- Four Temporal Values: $SI attributes contain a set of four temporal values.
- Core Metadata: The $SI attribute contains the core metadata and exists for all files and directories.

(2) Features

- Windows APIs: $SI attributes are used by the application programming interfaces (APIs) in the Microsoft Windows operating systems.
- Keeping Up to Date: This attribute will always be kept up to date. Any modification on digital files will update the date-time stamp values of $SI attributes. These attributes are those most frequently updated as a result of file activity.

2.1.2 $FN Attribute: Rarely Updated

(1) Contents

- Four Temporal Values: $FN attributes contain another set of four temporal values, which are not updated as often as their counterparts in the $SI attribute.
- File Name: This is where users store the file name and its parent directory's address.

(2) Features

- Rarely Updated: $FN attributes often correspond to the file creation time and are rarely updated.

- Updated Situation: They frequently correspond to when the file was created, moved, or renamed. Moving a file from one drive to another will update $FN to reflect the file created status in $SI attribute.
- Differ from $SI: $FN attributes contain many values that are duplicated with $SI, but its result of file activity differs from each other.
- Full Path: The address of parent directory can be used to determine the full path of file.

2.1.3 $DATA Attribute

Data can be hidden in files or directories as well. Codes can be executed directly from ADS. This makes ADS a covert vector for malware activities [2, 8, 9]. ADS is not dangerous itself but makes digital files or directories vulnerable to exploit by malicious hackers. Any information stored in the files has many potential advantages in digital forensics.

(1) Contents
- Raw Data of a File: ADS allows files to contain more than one stream of data. The default data stream is called '$DATA.'
- Command Line Process: Executables in ADS can be executed from the command line. Since the $DATA alternate stream exists for every file, ADS can be operated easily by echo, type, start, or other commands on the compromised machines for a rootkit or malware.
- Any Format: ADS can hide data in additional $DATA attributes as any formats, such as txt, doc, or jpg.
- Any Size: That additional $DATA attribute, allocated for each file, has any size of digital file.

(2) Features
- Metadata Update: Any modification of raw data will affect some EMAC values in $SI or $FN attributes.
- Names to Hide Data: $DATA holds the raw data of a file. Additional $DATA attributes can be allocated to an MFT entry, but they must have names to hide data.
- Special Toolkits to Locate ADS: ADS is not shown when the contents of a directory are listed. Special toolkits are necessary to locate it.

2.2 Temporal Values in NTFS

NTFS deals with every attribute as a file and maintains at least one MFT entry containing various attributes to store corresponding metadata and multiple date-time stamps [6]. Details for each attribute come from the $AttrDef of NTFS' hidden system files. $AttrDef is made up of multiple 160 byte records, and it contains attribute definitions [3]. The attributes of $SI and $FN in the NTFS $AttrDef file hold the following four forensically interesting values (EMAC for short) [3, 7]: MFT Entry modified (or metadata change) time (E-time), Last Modification Time (M-time), Last Access Time (A-time), and Creation time (C-time). $MFT is Windows Master File Table it stores metadata about the files on a system.

The $MFT is located in the Windows registry. EMAC times record pieces of file system metadata when certain events pertaining to a digital file occurred most recently. They are updated in different circumstance. Windows does not often update $FN temporal values when $SI attributes are much sensitive to the file's diverse processes. Four temporal attributes on $SI are explained below (Fig. 1) [2, 3]:

2.2.1 Entry Modified Time (E-time): The File Metadata Was Last Modified

The E-time is updated when any of MFT entry attributes are changed. It means that every modification of file will update this attribute.

2.2.2 Last Modification Time (M-time): The Content of the $DATA or $INDEX Attributes Was Last Modified

The M-time of $SI changes if the file's content or summary properties are modified. The M-time is updated if the file content, file summary properties or the value of any $DATA attributes are modified. However, if users modify the attributes or the name of a file, this value should remain the same.

2.2.3 Last Accessed Time (A-time): The File Metadata Was Last Accessed

A-time is updated when the metadata or content is viewed. Windows operating system does not update A-time by default.

2.2.4 Creation Time (C-time): The Time that the File Was Created

The C-time of $SI is created for a new file. This attribute will not be updated by any legal operations.

3 Experiment Design

The experiment design is intended to support the date-time stamp analysis of digital evidence in ADS behaviors. The objective is to identify significant operational events in file transfers across Microsoft NTFS. The challenge of these experiments lies in the hidden date-time stamps not easily visible and extracted using regular File Explore in Windows. These hidden date-time stamps contain more critical information about the file operations, which require rigorous procedures to extract.

3.1 Experiment Environment

To simplify this experiment, the file sizes in this paper are less than 1 MB. The research environment for the experiments is illustrated below.

- OS: Windows 7 Ultimate, 64-bit
- File System: NTFS file system
- Microsoft Office Suites: Word, PowerPoint and Excel

- Forensic Tool: Forensic Toolkit (FTK) 6.2.1 (http://accessdata.com/product-download/ftk-download-page)
- ADS tool: AlternateStreamView v1.53 (64-bit) (http://www.nirsoft.net/utils/alternate_data_streams.html)
- File Size: The file sizes in this experiment are less than 1 MB.

3.2 Experiment Observations

The research environment was created to test files and observe their different attributes on file metadata, timestamp, and other related issues. To reiterate, the experiment design is divided into three stages: date-time stamp observation on text or Word file, embedding txt file into the ADS of text/Word file, and embedding other files into the ADS of text/Word file.

3.2.1 First Phase: Date-Time Stamp Observation on Text or Word File

(1) **Observation in Baseline Environment**

Both Word format and text file format are set for the experiment. Date-time stamps are recorded on Sep. 11, 2017 after files are created (Table 1) or modified (Table 2) for comparison.

(2) **Time Rule Observation: Update Information on Temporal Attributes**

- File Creation

  When a text or word file is created, date-time stamps attributes are created.

  – Rule 1a: File Creation on Text File

  In Rule 1a, the attributes of \$SI and \$FN in the MFT are the same.

$$\$SI.EMAC\text{-}time\ = \$FN.EMAC\text{-}time\,(Rule\ 1a)$$

  – Rule 2a: File Creation on Word File

  The attributes of \$SI and \$FN in the MFT are similar to each other and C-time is earlier than EMA-time. It takes limited time to process the Word file. If there is time delay, it can still be inferred to be the same event in Rule 2a. EMA-time is updated after its file creation. Time delay differs from the different file sizes.

$$C\text{-}time \leqq EMA\text{-}time\,(Rule\ 2a)$$

- File Modification

  – Rule 1b: File Modification on Text File

  The modification of text file will only update \$SI.ME-time (Table 1).

$$\$SI.AC\text{-}time \leqq \$SI.EM\text{-}time\,(Rule\ 1b)$$

  – Rule 2b: File Modification on Word File

  Microsoft Office is a complicated application for somebody who wants to create, modify, or embed data [8]. When a Word is modified, both \$SI.EMA-time and \$FN.EMA-time are updated (Table 2). Similar situations exist in other MS office files, such as Excel or PowerPoint. The different part in the follow-up Tables is underlined for comparison.

$$C\text{-}time \;<\; EMA\text{-}time\,(Rule\,2b)$$

Table 1. Date-time stamp observation on text file

Rule	Operation	$SI[a]				$FN[b]			
		E	M	A	C	E	M	A	C
1a	Text file creation	1	1	1	1	1	1	1	1
1b	Text file modification	2[c]	2[c]	1	1	1	1	1	1

[a] In $SI, AC-time ≦ EM-time.

[b] In $FN, EMAC-time keeps unchanged.

[c] The different part is underlined for comparison.

Table 2. Date-time stamp observation on Word file

Rule	Operation	$SI				$FN			
		E	M	A	C	E	M	A	C
2a	Word file creation[a]	1	1	1	1	1	1	1	1
		1′	1′	1′	1	1′	1′	1′	1
2b	Word file modification[b]	2	2	2	1	2	2	2	1

[a] In $SI, C-time ≦ EMA-time.

[b] There is time delay to process the Word file due to file sizes.

3.2.2 Second Phase: Embedding Txt File into the ADS of Text/Word File

Date-time stamps are recorded after ADS is embedded.

(1) **Observation in Experiment Environment**

- File Creation
  To understand the ADS influence on date-time stamps of target file, Word file (test.docx in E1-1) and text file (test.txt in E1-2) in Table 3 are created for comparison.
- Embed Data by ADS Hiding
  To understand how the ADS has influenced on the target file (Word file in E2-1 and text file in E2-2), date-time stamps of these files are observed in Table 3.

(2) **Time Rule Observation**

Any creation or change for embedding ADS hiding data into text or Word file meets the rule 2 of text file modification. The attribute of $SI.ME-time is only updated. Other attributes keep unchanged.

Table 3. Date-time stamp observation on ADS attributes

Time rule	Operation	Date-time stamp (+0000)	$SI[a]				$FN[b]			
			E	M	A	C	E	M	A	C
2a	E1-1: Word file creation	05:03:09(1)/ 05:03:10(1′)	1′	1′	1′	1	1′	1′	1′	1
1a	E1-2: Text file creation	05:03:20 (2)	2	2	2	2	2	2	2	2
1b	E2-1: Embed data (text file) into the ADS of word file	05:10:46(3)	3	3	1′	1′	1′	1′	1′	1
1b	E2-2: Embed data (text file) into the ADS of text file	05:11:41(4)	4	4	2	2	2	2	2	2

[a] In $SI, C-time ≦ A-time ≦ EM-time.
[b] In $FN, EMAC-time keeps unchanged.

3.2.3 Third Phase: Embedding Other Files into the ADS of Text/Word File

Embedding other files into the ADS of text/ˋWord file is recorded in Table 4.

(1) **Observation in Experiment Environment**

- Operation 1: File Decompression
  In E3-1–E3-2, File decompression is similar to file creation. But $SI.M-time keeps unchanged.
- Operation 2: ADS Embedding
  ADS can hide a file inside another file. In E4-1–E7-2, embedding data, such as txt, docx, exe, or jpg, into the ADS of text (or Word) file is similar to the modification rule of text file (Rule 1b).
- Operation 3: ADS Extraction
  In E8-1 and E8-2, all EMAC-time keeps unchanged when users extract ADS by AlternateStreamView software.
- Operation 4: ADS Deletion
  In E9-1 and E9-2, E-time is updated when ADS is deleted from text/Word file.

(2) **Time Rule Observation**

- Rule 3a: File Decompression
  File decompression is similar to file creation. But the $SI.M-time keeps unchanged. It means that the content of Word file is still the same.

$$\$SI.M\text{-}time\ <\ \$SI.EAC\text{-}time \ldots (Rule\ 3a)$$

- Rule 3b: ADS Deletion
  In $SI, E-time is updated when ADS is deleted.

$$\$SI.MAC\text{-}time\ <\ \$SI.E\text{-}time \ldots (Rule\ 3b)$$

Table 4. Date-time stamp observation on ADS attributes

Rule	Operation	Date-time stamp (+0000)	$SI[a]				$FN[b]			
			E	M	A	C	E	M	A	C
3a	E3-1: Decompress Word file	05:03:10(1)/ 09:44:27(1″)	1″	1[c]	1″	1″	1″	1″	1″	1″
3a	E3-2: Decompress text file	05:03:20 (2)	2	1[c]	2	2	2	2	2	2
1b	E4-1: Embed data (text file) into the ADS of Word file	09:50:26(3)	3	3	1″	1″	1″	1″	1″	1″
1b	E4-2: Embed data (text file) into the ADS of text file	09:50:58(4)	4	4	1″	1″	1″	1″	1″	1″
1b	E5-1: Embed data (docx file) into the ADS of Word file	09:52:23(5)	5	5	1″	1″	1″	1″	1″	1″
1b	E5-2: Embed data (docx file) into the ADS of text file	09:52:57(6)	6	6	1″	1″	1″	1″	1″	1″
1b	E6-1: Embed data (exe file) into the ADS of Word file	09:54:36(7)	7	7	1″	1″	1″	1″	1″	1″
1b	E6-2: Embed data (exe file) into the ADS of text file	09:55:10(8)	8	8	1″	1″	1″	1″	1″	1″
1b	E7-1: Embed data (jpg file) into the ADS of Word file	09:56:28(9)	9	9	1″	1″	1″	1″	1″	1″
1b	E7-2: Embed data (jpg file) into the ADS of text file	09:56:59(10)	10	10	1″	1″	1″	1″	1″	1″
N/A	E8-1: Extract the ADS from Word file	10:07:42(11)	9	9	1″	1″	1″	1″	1″	1″
N/A	E8-2: Extract the ADS from text file	10:08:01(12)	10	10	1″	1″	1″	1″	1″	1″
3b	E9-1: delete the ADS from Word file	10:09:00(13)	13	9	1″	1″	1″	1″	1″	1″
3b	E9-2: delete the ADS from text file	10:09:07(14)	14	10	1″	1″	1″	1″	1″	1″

[a] In $SI, AC-time ≦ EM-time.

[b] In $FN, EMAC-time keeps unchanged.

[c] $SI.M-time keeps unchanged in decompressing files.

3.3 Experiment Findings on Temporal Patterns

The date-time stamps line up with previous NTFS artifacts, and give investigators an idea of when the file system was created. The contributions of this paper are to establish date-time stamp patterns for their creation and its follow-up operations. To assist the reconstruction of events through the analysis of EMAC-time, the following phenomena are observed and elaborated.

Pattern 1: Different Update Patterns of Date-Time Stamps on File Types
Different file types have different update patterns on date-time stamps. $SI attributes are used by the APIs in the Microsoft Windows operating systems. For example, $SI.M-time is updated in Word file modification but keep unchanged in text file modification (Tables 1 and 2).

Pattern 2: Mutual Comparison between $SI and $FN
The EMAC-time should be treated as circumstantial unless they are verified via other data or information. The $SI information may be unreliable since some utilities can change the $SI information easily. The $FN attributes can be used to question the accuracy of the $SI attributes since they are not updated so often.

Pattern 3: Putting these Time Rules All Together
In Table 5, a new time rule A is concluded for all operations in this experiment. $SI.E-time refers to when the MFT entry for that file was last change. As the MFT entry contains a lot of metadata information about the file, including, size, name, location on the disk, parent folder, and creation date, changing any one of these should also change the E-time. It means that $SI.E-time will update when there are renaming the file, moving the file into a different folder, or increasing the file size. Every modification of file will update SI.E-time attribute, which is more reliable than others.

Table 5. Reliable temporal patterns for SI.E-time attribute

Time rule	Operation	Inequality	Finding
1a	Text file creation	$SI.EMAC-time = $FN.EMAC-time	$SI.MAC-time ≦$SI.E-time (Rule 4)
1b	Text file modification	$SI.AC-time ≦ $SI.EM-time	
2a	Word file creation	C-time ≦ EMA-time	
2b	Word file modification	C-time < EMA-time	
3a	File decompression	$SI.M-time < $SI.EAC-time	
3b	ADS deletion	$SI.MAC-time < $SI.E-time	

- Rule 4: All Operations in this Experiment

$$\$SI.MAC\text{-}time \leqq \$SI.E\text{-}time\ (Rule\ 4)$$

Pattern 4: File Size and Time Delay

A date-time stamp is a sequence of encoded information, which identify when a certain event occurred or a document was received. It is typically a record of the date and time of an action. A small delay is acceptable in the computer world [7]. If the file sizes in this experiment are large, then the time delay will become much serious. Future research will experiment more on large file sizes and observe their differences from the above findings.

4 Conclusions

Pattern finding may or may not play a role in crime reconstruction, but the progressive application of ADS experiment is evident. This paper introduces ADS and reviews the literature pertaining to the application of its data hiding to digital investigations and forensics. It describes some patterns for evaluating if ADS are hidden in Word or not. The application of file metadata and ADS analysis assists in accurately correlating activities from date-time stamp evidence. The utility of these patterns can be applied in forensic investigation. The contributions of this paper are to establish date-time stamp patterns for text/Word files and their follow-up ADS embedding.

Acknowledgment. This research was partially supported by the Executive Yuan of the Republic of China under the Grants Forward-looking Infrastructure Development Program (Digital Infrastructure-Information Security Project-107) and the Ministry of Science and Technology of the Republic of China under the Grants MOST 106-2221-E-015-002-.

References

1. Arnes, A.: Digital Forensics, pp. 147–190. Wiley, Hoboken (2017)
2. Carrier, B.: File System Forensic Analysis, pp. 273–396. Pearson Education Inc., London (2005)
3. Casey, E.: Handbook of Digital Forensics and Investigation, pp. 209–300. Elsevier Inc., Amsterdam (2010)
4. Casey, E.: Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet, 3rd edn., pp. 187–306. Elsevier Inc., Amsterdam (2011)
5. Chow, K.P., Law, F.Y.W., Kwan, M.Y.K., Lai, K.Y.: The rules of time on NTFS file system. In: 2nd International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE), Bell Harbor, WA, USA, 10–12 April 2007
6. Ding, X., Zou, H.: Reliable Time Based Forensics in NTFS, pp. 1–2. School of Software, Shanghai Jiao Tong University (2010)
7. Kao, D.Y.: Cybercrime investigation countermeasure using created-accessed-modified model in cloud computing environments. J. Supercomput. Spec. Issue Emerg. Platf. Technol. 1–20 (2015)
8. Krahl, K.M.: Using Microsoft Word to Hide Data. Thesis, pp. 1–13. Utica College, ProQuest Dissertations Publishing (2017)
9. Mahajan, R.: Design and Development of Improved Stealth Alternate Data Streams. Thesis, pp. 6–42. Thapar University, Patiala, India (2014)

Ant-Based Botnet C&C Server Traceback

Chia-Mei Chen[1](✉) and Gu-Hsin Lai[2]

[1] Department of Information Management, National Sun Yat-sen University,
Kaohsiung, Taiwan
cchen@mail.nsysu.edu.tw

[2] Department of Technology Crime Investigation, Taiwan Police College,
Taipei, Taiwan

Abstract. Botnets can cause significant security threat and huge loss to organizations, and are difficult to discover their existence; therefore they have become one of the most severe threats on the Internet. The core component of botnets is their command and control server (C2 server or C&C server) through which the bot herder instructs zombie machines to launch attacks. A commonly used protocol, such as IRC (Internet Relay Chat) or HTTP, is adopted to communicate between bot ma-chines and the server. In addition, some advanced botnets might have multiple C2 servers to evade detection and to extend the life time. Therefore, identifying the C2 server is important to prevent botnet attacks or further damage. In this paper, detection scheme based on ant colony optimization algorithm is proposed to identify the paths from bot machines to the C2 server. The results show that the proposed detection can identify botnet servers efficiently.

Keywords: Botnet · Anomaly detection · Ant colony optimization

1 Introduction

A bot is an automated software performing operations instructed by the botmaster by means of a command and control (C&C or C2) server. Botmaster infects hosts through various attacks, such as malicious web pages, spam mails, viruses, or worms. Bots are increasingly used for malicious purposes. An estimated one million PCs are under the control of hackers worldwide. These botnets ranged in size from a few hundred compromised PCs to 50,000 machines [15].

Botmaster builds a C&C server using a commonly used network protocol, hiding and blending malicious transmissions in a vast amount of normal user traffic. This makes botnet detection challenging. C&C server plays a vital role in a botnet, as it contains the information of the bot machines and controls the malicious operations such as DDoS attacks [12]. According to the botnet architecture, a botnet can be taken down accordingly as long as the law enforcement shuts down the botnet server. Therefore, to evade the detection of botnet servers, advance botnets adopt fast-flux domain technology to extend the lifetime and robustness of botnet [10, 11, 17, 19, 21]. Traditional network intrusion detection systems (NIDS) fail to identify botnet servers in a network.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 286–292, 2018.
https://doi.org/10.1007/978-3-319-76451-1_27

Even though the malicious traffic is small, the communication exhibits some anomaly behavior as a bot is robot software. Normal user requests are issued at a random time and the contents are diverse, while bots may connect to the server periodically and the message content may be limited. This study proposes an ant-based detection mechanism to identify the botnet servers which have the above anomalous traffic with the client machines in a network. Ant colony optimization (ACO) finds an optimal path based on pheromone. Pheromone proposed in this study is the degree of anomaly of a traffic flow. Therefore, the ant algorithm can be adopted to find the anomalous traffic between bots and the C&C servers and hence to detect the C&C servers.

2 Related Work

A botnet usually takes advantage of standard network protocols such as IRC or HTTP to remotely control victim terminals for spreading malware. The main reason for choosing HTTP is so that hackers can write control commands directly into the web program, which can easily allow a web-based botnet to be hidden inside the normal traffic flow so that it can remain undetected until the actual attack is launched [15, 16].

A botnet detection system BotGAD (Botnet Group Activity Detector) is developed, based on the group activity model and metric, including group uniformity, activity periodicity, and activity intensity [3]. Lakhina et al. adopted sample entropy to find the traffic flow distribution characteristics. The detection approach can detect various attacks such as DDoS and port scan during the progress of the attacks, but it is not able to identify botnets prior to the attacks [7].

AsSadhan et al. employed periodograms to study the periodic behavior of botnet and monitor the command and control communication traffic [2]. The Walker's large sample test is applied to detect the C&C traffic whether bot traffic is or not. Yen and Reiter proposed a detection system called TAMD to identify infected hosts in the enterprise network by finding out aggregated communication involving multiple internal hosts [18]. The aggregated features include flows communicating with the same external network, sharing similar payload, and involving internal hosts with similar software platforms. The experimental results show that the proposed approach has a low false positive rate.

A bot is a program which can perform a fixed number of instructions. All bots commit malicious activities according to the botmaster's commands. Akiyama et al. proposed three metrics for determining the botnet behaviors: relationship, response, and synchronization [1]. The relationship presents the connection between botmaster and bots over one protocol, such as IRC, HTTP, or P2P. The response means that bots respond immediately and accurately after they receive commands from the botmaster. The synchronization means bots simultaneously carry out programmed activities, such as DDoS attack, reporting their status, or sharing information, based on the botmaster's commands.

ACO has been applied to shortest path routing, traveling salesperson [4, 20], and optimal network routing problems. The network routing research [14] demonstrated that ACO performs better than others and introduced two types of ants for changing routing cost: regular ant and uniform ant. Regular ant selects its path based on the amount of pheromone, while the path chosen by uniform one is based on user's choice. Such approach can avoid regular ant from exploring to a local optimal for it cannot remove or add a node adaptively. The improved ACO is more suitable for finding an optimal routing in dynamic network environments.

The above research inspired us to apply ACO to C&C server detection in a dynamic botnet communication environment, while the communication contains some anomaly behaviors helpful for ants to select the paths with anomalies.

3 Ant-Based C&C Server Detection

Some research focuses on botnet path tracking such as [16], assuming that the traffic information on the edge routers of the inter-connected network can be obtained, while, in reality, each network is autonomous and does not share traffic flow information with other network domains. A more practical solution for a network administrator to identify if its network contains anomalous bot traffic to C&C server is to examine its own network traffic information and to identify suspicious botnet servers. The proposed ant-based botnet server detection is for such purpose. Therefore, only outbound traffic of a network is examined.

In this paper, an IRC traffic logger, IRC sniffer, is deployed to collect IRC traffic flows in the corporate network. The payload information is extracted from payload of IRC traffic including IRC commands such as JOIN, USER, PASS and IRC messages embedded in PRIVMSG. In this paper, the basic analysis unit, a flow, is defined as 6-tuple R = {Sip, Dip, Sport, Dport, Time, Payload}.

This study adopts three attributes to define the anomalous communication between bots and its server: flow regularity, content similarity, and keyword similarity. The following notations will be used for computing the attributes. Let $CR(Sip_a, all) = CR_a$ be the set of traffic flows from Sip_a, $CR(Sip_a, Dip_b) = CR_{ab}$ be the set of traffic flows from Sip_a to Dip_b, and $R_i(Sip_a, Dip_b)$ be the ith flow from Sip_a to Dip_b. $|CR_{ab}|$ denotes the number of traffic flows in CR_{ab} and $|R_i|$ denotes the size of flow R_i.

The attribute flow regularity contains three indices: S_t: the standard deviation of the interleave time of two consecutive connections, S_s: the standard deviation of the packet size, and h: the ratio of the number of traffic flows destined to Dip_b to the total number of flows in the traffic cluster $CR(Sip_a, all)$.

The second attribute, content similarity, compares the message content similarity by longest common subsequence (LCS) and averages the degree of the similarity of all traffic flows in a given time frame.

The third attribute identifies the suspicious keywords in the messages. As a bot machine is not a human, it understands a limited set of commands or words. Therefore, the third attribute computes the average ratio of the number of keywords appeared in the messages between the two parties.

Ant-Based Detection Algorithm

An isolated ant moves essentially random. It decides to follow a trail with high pheromone trail and reinforces the trail by laying its own pheromone. The collective behavior emerging from ants is a form of autocatalytic reaction where the more the ants follow a trail, the more attractive the trail becomes. The proposed ant-based detection algorithm develops heuristic information, anomaly score function, which signifies the immediate impact that a local decision might have on solution quality. For example, in Traveling Salesperson Problem, the heuristic information is inversely proportional to the distance. In this study, the heuristic information indicates the degree of traffic anomaly observed on a path. Therefore, traffic path exhibiting bot-server connection behaviors has high pheromone and more ants will explore such path. If the same traffic path continues showing such anomaly, ACO will form a positive feedback and finally most ants will explore the same path.

In the initialization phase, a group of ants is positioned on a client machine in the network. The cluster of traffic flows from the client Sip_a in a given time frame, i.e., $CR(Sip_a, all) = CR_a$ is examined.

ACO has been applied for shortest path problems in the literature. In the pheromone calculation, visibility function, η, is often defined as the reciprocal of distance, where the shorter distance contributes larger visibility and results in shorter distance path. In this study, the visibility function defined by the proposed anomaly score function indicates the degree of the anomaly of the network traffic. More anomalous traffic results in high anomaly score and then higher pheromone.

The intensity of the pheromone is updated after each cycle of path exploration. A portion of the current pheromone will be evaporated and more pheromone will be accumulated, if the path is explored in the next time frame.

The accumulated pheromone is to sum up the pheromone laid by the ants exploring the path during the next time frame.

Each time when all ants complete one iteration (cycle), the intensity of the pheromone on each path will be recalculated based on the above equations. The ant-based detection scheme iterates until the tour counter reaches the pre-defined number of cycles. Once the traffic of all client machines in the network have been explored by ants, the amount of the pheromone collected by each destination from multiple sources is summed up, which represents the anomaly degree of the destination. The suspicious botnet servers are the ones with high pheromone.

4 Performance Evaluation

The performance evaluation is to evaluate the detection performance of the proposed detection system under various network environments.

The experiments were implemented on a testbed, building simulated networks which consist of the following machines: botmaster, a number of C&C servers, a number of bot and normal machines, a victim, and a traffic collector. The C&C servers were implemented using open source IRC server Unreal IRCd [24]; the botmaster applied a popular IRC software mIRC [23] to control the botnet; the traffic collector was implemented based on Wireshark [25] for monitoring and re-cording the network traffic.

This experiment is to evaluate the detection performance of the proposed method under various infection rates in a network. The literature has demonstrated that advanced botnets might contain multiple C&C servers to increase the stealth and resilience. The simulation network environments are illustrated in Fig. 1.

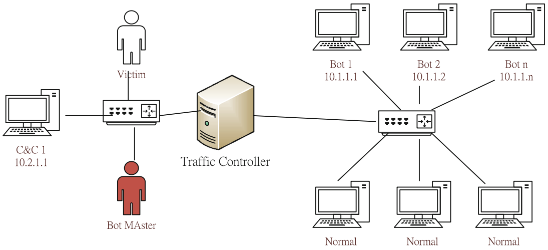


Fig. 1. Experimental environment.

To observe if the proposed detection algorithm can identify the C&C servers and bots efficiently, all the experiments were blended in various amounts of malicious and normal (including peer-to-peer) traffic. To observe if the proposed ant-based detection system can identify bots and the C&C server and bots in a network with very few number of infections and little amount of malicious traffic. The experimental results are shown in Table 1 and demonstrate that the proposed system could detect both C&C servers and bots effectively.

Table 1. Detection performance for C&C server.

Bots:Normal	Traffic ratio of M:N:P	No. of malicious servers	No. of suspicious servers (detected)
1:9	3:97:0	1	1
5:5	3:97:0	1	1
10:0	3:7:0	1	1

5 Conclusion

Botnet communication becomes stealthy to evade rule-based intrusion detection system, where a small amount of malicious traffic is generated and mixed into mass amount of normal traffic. This study develops a novel visibility function of the ant colony optimization algorithm based on the traffic anomaly; therefore, the paths to malicious servers receive high pheromones.

The proposed ant-based detection system requires no priori information of the whole network topology or the flow information of other routers of the whole network and could identify malicious C&C servers in the early stage of botnet infection with a small amount of malicious traffic.

The proposed solution is evaluated on simulated network environments with various mixture of malicious and normal traffic. More evaluations can be done using real botnet traffic collected from a large real network. Further investigation can be done by extending to peer-to-peer botnets.

Acknowledgments. The study is based on the work sponsored by the Ministry of Science and Technology under the grant MOST 106-2221-E-110-017-MY3.

References

1. Akiyama, M., Kawamoto, T., Shimamura, M., Yokoyama, T., Kadobayashi, Y., Yamaguchi, S.: A proposal of metrics for botnet detection based on its cooperative behavior. In: SAINT Workshops, p. 82 (2007)
2. AsSadhan, B., Moura, J.M.F., Lapsley, D.E.: Periodic behavior in botnet command and control channels traffic. In: GLOBECOM, pp. 1–6 (2009)
3. Choi, H., Lee, H., Kim, H.: BotGAD: detecting botnets by capturing group activities in network traffic. In: Proceedings of the Fourth International ICST Conference on Communication System Software and Middleware (2009)
4. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system: optimization by a colony of cooperating agents. J. IEEE Trans. Syst. **26**(1), 1–13 (1996)
5. Kondo, S., Sato, N.: Botnet traffic detection techniques by C&C session classification using SVM. In: International Workshop on Security, pp. 91–104 (2007)
6. Lai, G.H., Chen, C.M., Jeng, B.C., Chao, W.: Ant-based IP traceback. Exp. Syst. Appl. **34**(4), 3071–3080 (2008)
7. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: SIGCOMM, pp. 217–228 (2005)
8. Livadas, C., Walsh, R., Lapsley, D.E., Strayer, W.T.: Using machine learning techniques to identify botnet traffic. In: 31st IEEE Conference on Local Computer Networks, pp. 967–974 (2006)
9. Lu, W., Rammidi, G., Ghorbani, A.A.: Clustering botnet communication traffic based on n-gram feature selection. In: Proceedings of Computer Communications, pp. 502–514 (2011)
10. McGrath, D.K., Kalafut, A.J., Gupta, M.: Phishing infrastructure fluxes all the way. IEEE Secur. Priv. Mag. **7**(5), 21–28 (2009)
11. Nazario, J., Holz, T.: As the net churns: fast-flux botnet observations. In: Proceedings of the 3th International Malicious and Unwanted Software (Malware), pp. 24–31 (2008)
12. Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., Knightly, E.: DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. IEEE/ACM Trans. Networking **17**(1), 26–39 (2009)
13. Strayer, W.T., Walsh, R., Livadas, C., Lapsley, D.E.: Detecting botnets with tight command and control. In: 31st IEEE Conference on Local Computer Networks, pp. 95–202 (2006)
14. Subramanian, D., Druschel, P., Chen, J.: Ants and reinforcement learning: a case study in routing in dynamic networks. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 832–839 (1997)
15. Trend Micro, Botnet threats and solutions: phishing (2006). http://anti-phishing.org/sponsors_technical_papers/trendMicro_Phishing.pdf
16. Wang, P., Lin, H.T., Wang, T.S.: A revised ant colony optimization scheme for discovering attack paths of botnet. In: IEEE International Conference on Parallel and Distributed Systems, pp. 918–923 (2011)

17. Wu, J., Zhang, L., Liang, J., Qu, S., Ni, Z.: A comparative study for fast-flux service networks detection. In: Sixth International Conference on Networked Computing and Advanced Information Management, pp. 346–350 (2010)
18. Yen, T.F., Reiter, M.K.: Traffic aggregation for malware detection. In: Lecture Notes in Computer Science, pp. 207–227 (2008)
19. Zhu, Z., Lu, G., Chen, Y., Fu, Z., Roberts, P., Han, K.: Botnet research survey. In: 32nd Annual IEEE International Conference in Computer Software and Application, pp. 967–972 (2008)
20. Upton, G.: An ant colony optimization algorithm for the stable roommates (2002). http://www.cs.earlham.edu/~uptongl/project/senior_thesis.html
21. Huang, C.Y.: Effective bot host detection based on network failure models. Comput. Netw. **57**(2), 514–525 (2013)
22. Testbed@TWISC. http://testbed.ncku.edu.tw/
23. mIRC. http://www.mirc.com/
24. Unreal IRCd. http://www.unrealircd.com/
25. Wireshark. http://www.wireshark.org/

Public Key Systems and Data Processing

T-Brain: A Collaboration Platform for Data Scientists

Chao-Chun Yeh[1,3](✉), Sheng-An Chang[3], Yi-Chin Chu[3], Xuan-Yi Lin[3], Yichiao Sun[3], Jiazheng Zhou[3], and Shih-Kun Huang[1,2]

[1] Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan
skhuang@cs.nctu.edu.tw

[2] Information Technology Service Center, National Chiao Tung University, Hsinchu 300, Taiwan

[3] Computational Intelligence Technology Center, Industrial Technology Research Institute, National Chiao Tung University, Hsinchu 300, Taiwan
{avainyeh,madchang,nelson.chu,xylin,icsun,zhou}@itri.org.tw

Abstract. When data were generated easily and rapidly with mobile services and computing power can increase on demand with the cloud computation service, data scientists who work with huge data can solve challenging problems. Smart intelligent applications such as Go, healthcare and self-driving vehicles show great improvement recently. In addition to those problems, there are still more complex problem such as weather impacts analysis, financial crisis prediction and crime prevention and so on. To overcome those challenging problems, many crossdisciplinarity or interdisciplinary experts have to collaborate for the solutions. In the paper, we propose a collaboration platform and a system design for data scientists to share data, write analytic scripts and discuss topics related with those problems. In current status, eleven dataset were collect ed such as spam mail, malware data, honeynet log, Hadoop workload log and some other open data and based on those dataset and improvement local cache design (i.e., average response time improvement 92.36% and request availability improvement 70%). With the platform, many education and competition activities can be hold successfully on the collaboration platform.

Keywords: Collaboration platform · Data scientist · Docker · Jupyter

1 Introduction

When data were generated easily and rapidly with the mobile services and computing power can increase on demand with the cloud computation service, data scientists who work with huge data can solve challenging problems. For example, Go [1], healthcare [2] and self-driving vehicles [3] show great improvement recently. In addition to those problems, there are still more complex problem such as weather impacts analysis, financial crisis prediction and crime prevention and so on. To overcome those challenging problems, many crossdisciplinarity or interdisciplinary experts have to collaborate for solutions. In the paper, we propose a collaboration platform for data scientists to share data, write analytic scripts and discuss topics related to those problems.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 295–309, 2018.
https://doi.org/10.1007/978-3-319-76451-1_28

2 Background

2.1 Collaboration Platform

A collaboration platform is designed to help knowledge workers involved in a typical task to achieve their goals (e.g., the solution or the employee). There are many collaboration platforms for different purposes such as innovation based platforms, data driven based platforms and system based platforms. The innovation based platform (i.e., Innocentive [4]) focus on creative idea and methodology for cost-effective solution. The data driven based platform [5–7] targets on the solution with data. For example, the data providers such as companies and research centers post data and analytical problems. On the other hand, data scientists produce best models to describe and predict behaviors from the data. The system based platform [8, 9] provides market with software system or component for companies and engineers.

2.2 Docker and Docker Swarm

Docker [10] is an open-source container engine which provides isolated environment for running packaged applications. These isolated environments are called containers. A container usually packages an application with all its dependencies, making it self-contained and portable. This isolation and security layer allows a user to run multiple containers on a given host simultaneously, each with its own kernel-level namespace and network stack, without interfering each other. A computer host running Docker engine and its associated containers is called a Docker host.

Docker Swarm [11] is a native clustering for Docker. Docker Swarm groups a pool of Docker hosts and provides a single accessing interface for this group of Docker host, making it a bigger virtual Docker host. A Docker Swarm cluster can be used as a solution to run a group of containers which requires a set of resource, such as CPU and RAM, exceeding the capacity of a single Docker host. For example, if a system operator plans to run 64 containers, each requesting for two gigabytes of RAM, he can either run these containers on a single Docker host with more than 128 GB RAM or a Docker Swarm cluster consisting four Docker hosts with 32 GB RAM in each.

2.3 HDFS

Hadoop Distributed File System (HDFS) is a well-known distributed file system based on Google File System (GFS) [12] and designed to run on large clusters (i.e., thousands of computers) of small computer machines in a reliable, fault-tolerant manner. HDFS uses a master/slave architecture. There are two primary components at the HDFS: the first one is NameNode that manages namespace the file system metadata. The other is DataNodes that store the actual data in physical nodes. An HDFS file is split into three blocks, and these blocks are stored in a set of DataNodes for data recovery and access performance. Dataset of T-Brain has the write-once and read many times characteristic, and it is suitable for the HDFS properties. It can reduce conflicts in the concurrent control, improve the throughput of data accesses and support large datasets and files into highly the fault tolerant.

2.4 Jupyter/Jupyter Hub

Jupyter [13] is a language independent and open source interactive computing framework. For developers, it supports more than forty languages (e.g., Julia, Python and R) for programmers to develop with browsers. It defines the network protocol for interactive computing and client-side representation (i.e., Notebook and Markdown) and provides writing documents with plain text, equation editor and visualizations. Jupyter is widely used by data scientists for data cleaning, data transformation, numerical simulation, statistical modeling and machine learning. Those work can be condensed as Notebook format for sharing to achieve reproducible researches [14].

JupyterHub [15] extends Jupyter for multi-users authentication and spawning the Notebook instance to each user and RESTful API [16] for those functions. The sub-system of JupyterHub are Proxy, Hub and Single-user Notebook server.

Proxy: With the component of node-http-proxy, proxy was generated dynamically to route the http-requests to Hub and single user Notebook servers.

Hub: Functions of Hub are user account management, authentication and coordination with Single user Jupyter Notebook by Spawner.

Single-User Notebook Server: When user login, the Spawner allocates resource for Single-user Notebook server which provides user computing resource.

3 System Design and Implementation

3.1 System Architecture Overview

The T-Brain is developed on four physical machines with 16 cores (2.4 HHz/core), 64 GB RAM and 8 TB SATA totally and the software are Ubuntu 16.04 64-bit, Django 1.8.7, MariaDB 10.1, Hadoop-2.6.0 and JupyterHub 0.7.2. Figure 1 shows the architecture of T-Brain that includes three main sub-system below:

- T-Brain API
  T-Brain API includes modules such as user management, dataset, script, forum and activity. Those modules cooperate with each other to support specific tasks (e.g., data upload, analysis script editing/running and discussion).
- Distributed Storage
  Distributed Storage leverages Hadoop HDFS with write-once, read-many-times properties to save the uploaded data and scripts. The sub-system is controlled by Dataset module in T-Brain API.
- Computing Resource Pool
  Computing Resource Pool provides multiple docker instances with Jupyter web interface. Those instances are managed by JupyterHub and Docker swarm. We reference Hamrick's work [17] for the deployment of JupyterHub.

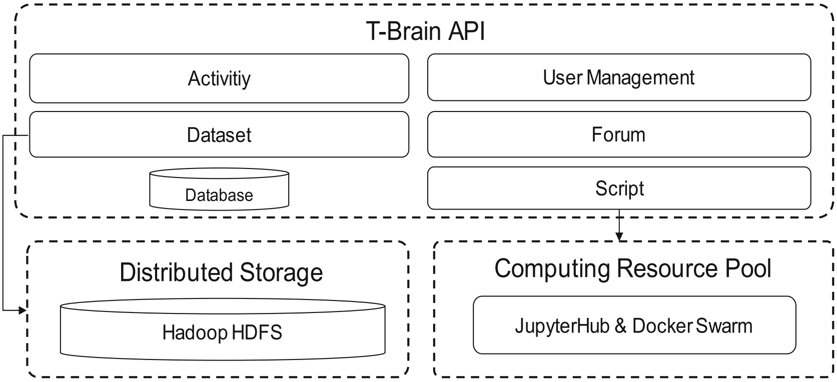


Fig. 1. T-Brain architecture including three main sub systems: (1) T-Brain API (2) Distributed Storage (3) Computing Resource Pool

3.2 User Management

User Management module affords basic authentication and authorization functions and maintains use profile. User Management module provides the following functions through RESTful APIs:

- Authenticate and authorize user.
- Provide end-user license agreements information and check the status.
- Set, query and update the user group.
- Set, query and update the user information.

3.3 Dataset

The Dataset module provides a space to store user-uploaded datasets. The data processing pattern on T-Brain is a write-once and read-many-times pattern, which prohibits modifications to existing datasets, to guarantee data consistency for long-running analytic jobs. User-uploaded dataset files are stored in HDFS. Dataset module includes file version control. A new version of a file is created whenever there is a file update obligation. Dataset module provides the following functions through RESTful APIs:

- Query dataset information with filtering and ordering.
- Create dataset and new dataset version.
- Modify dataset metadata.
- Upload dataset files to HDFS.
- Download dataset files.
- Publish/Un-publish a dataset.
- Put a dataset into maintenance mode.
- Check availability of a dataset.

Figure 2 shows the architecture of Dataset module. An asynchronous task queue is utilized to mitigate the long-blocking time for large dataset uploads. The asynchronous task consists of two main components: Redis [18] service and Celery [19] worker. Redis service acts as a message broker to collect file-uploading tasks for each dataset. The message broker maintains a task queue for all requests. Each request for uploading a file will be packaged into a Celery task and delivered to the message broker. Celery worker is responsible for actual execution of file-uploading tasks in the task queue maintained by Redis service. There are multiple Celery worker processes executing multiple tasks concurrently. Each Celery worker process keeps monitoring the message queue in the Redis service for pending tasks. A Celery worker fetches and executes a file-uploading task whenever it sees a pending file-uploading task in the message queue. The worker then compresses and uploads the dataset to HDFS and writes the execution result to MariaDB, after the task as been completed.

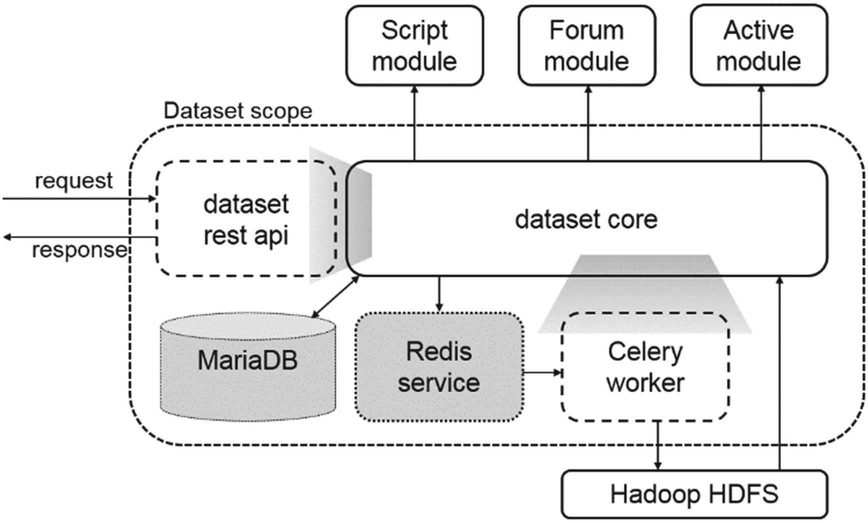


Fig. 2. T-Brain architecture of dataset module including dataset rest API, dataset core, MariaDB, Redis service and Celery worker

This design of an asynchronous task queue enables background uploading of datasets to HDFS. Therefore, user requests for uploading datasets can be returned sooner. Both Celery worker and Redis are scheduled to run at system boot time as system services. If one of the Celery worker process dies, the system will spawn a new Celery worker process to prevent from the Celery service interruption.

We evaluated three alternatives, namely libhdfs, snakebite and libhdfs3, for the implementation of HDFS access functions. We benchmark these three implementations by uploading and downloading files of different size and type. Figures 3 and 4 show the benchmarking results. File sizes for test files are: CSV file (519 MB); binary file (500 MB); SQLite file (1.2 GB). Errors were encountered when downloading file through libhdfs and the snakebite does not implement *put()* function so it cannot handle uploading tasks. As a result, libhdfs3 is selected because it not only handles both uploading and downloading but also performs better.

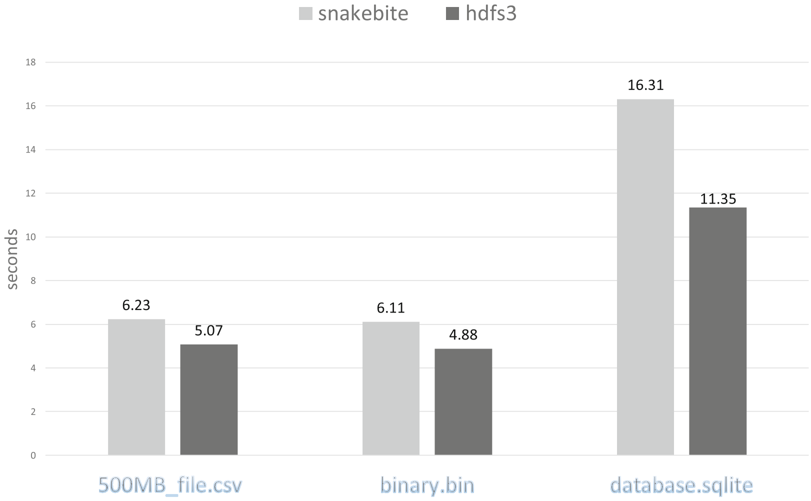


Fig. 3. Performance of download file from HDFS

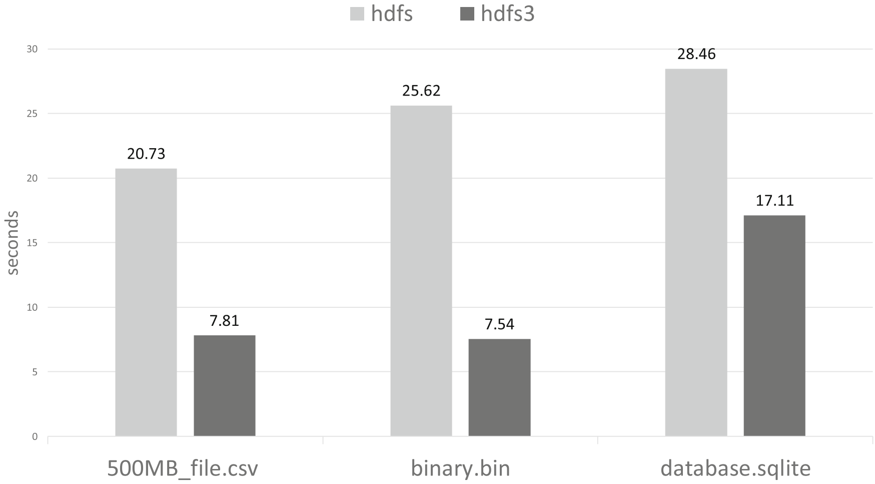


Fig. 4. Performance of upload file from HDFS

3.4 Script

Script module offers users to create their scripts for a chosen dataset in dataset module. For a script, user can open an editor to edit, execute, and publish it. The Script module has version control for the published scripts.

Script module integrates Jupyter Notebook to provide script editing and execution features. It offers RESTful APIs for website to query and control. There are three parts for the Script functions:

- Query script information.
- Open script editor and publish a script.
- Handle the network traffic from website to JupyterHub and Jupyter Notebook.

The script information and metadata are stored in relational database (i.e., MariaDB) and HDFS. When users access scripts, website will query the information through Script RESTful APIs. The following lists are the query functions:

- List scripts with filters and orders.
- Query the basic information about a script, such as creator, created time, and related dataset.
- Query the version of a script.
- Query the execution information about a script, including execution time, related Docker image, errors, and script size.
- Query the fork information about a script, including who forks the script, the forked time, and the new script title.

In T-Brain, opening a script is to open a Jupyter Notebook for editing, and user can perform analysis by the script execution. There are three modes to open a script:

- Create a new script for analysis.
- Create a new version based on an existing script.
- Fork a new script based on an existing script.

When user opens a script, and chooses a Jupyter Notebook template that are different supporting for different machine learning frameworks, Script module will perform the following operations. It will check status of Jupyter Notebook server through JupyterHub RESTful APIs. If status is ready, it will spawn a Docker container of Jupyter Notebook, record the Kernel Session ID, and download a corresponding dataset and a script from HDFS to the mounted volume. After performing analysis, user can publish the script through publish function. Script module will convert a script format (i.e., form IPYNB to HTML format) through notebook converter, store the file to HDFS, and record information in MariaDB, including script execution time, related Docker image, errors, and script size. At last, it closes the Jupyter Notebook by Kernel Session ID through RESTful API.

As shown in Fig. 5, for the first step, Script module helps to spawn Jupyter Notebook container. Since original JupyterHub does not offer authentication function for RESTful API, to connect from a website, the authentication part needs to be modified. Therefore, JupyterHub authentication is modified to provide two cookies to the front to access JupyterHub and Jupyter Notebook after T-Brain login. These cookies let users have enough permissions for operations, and they are jupyter-hub-token and jupyter-hub-token-[username] cookies for JupyterHub RESTful API and Jupyter Notebook RESTful API, respectively. Since Script module helps users to open Jupyter Notebook containers, users do not need to manipulate JupyterHub RESTful APIs. However, the cookie variable (i.e., jupyter-hub-token-[username]) must be set correctly for the Jupyter Notebook access.

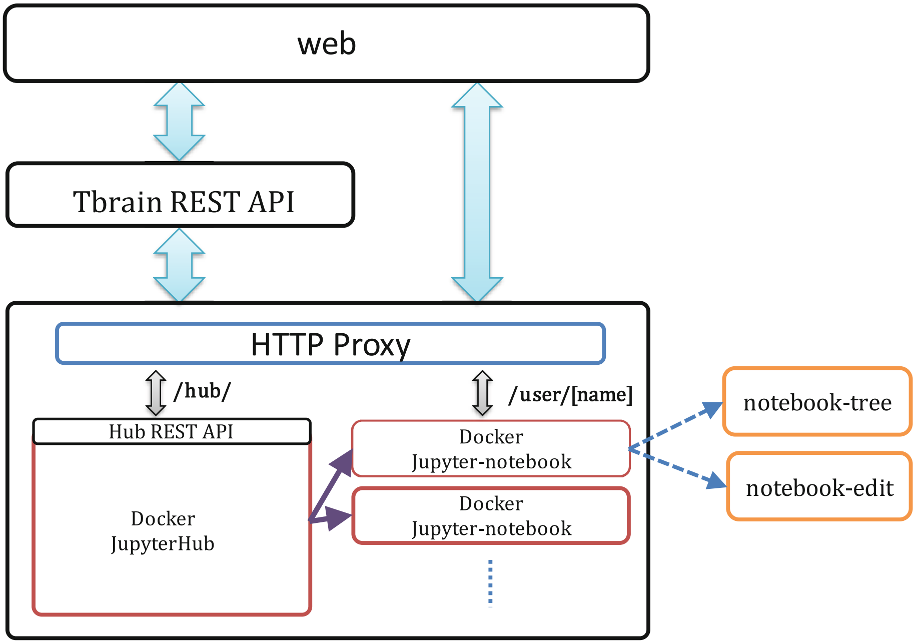


Fig. 5. The illustration of Script module

After a user is authenticated, Jupyter Notebook container is spawned, and the cookie variable is set, for the second step, Script module will route user to corresponding URL of Jupyter Notebook (/user/[username]) through a revert proxy. Moreover, Jupyter Notebook server incorporates with HTTPS and Web Socket at the same time, the related web header needs to be set accordingly as well.

3.5 Forum

Forum module supports discussion for data scientists from dataset and script and topics from the web interface. The post object is the basic unit in Forum module and it includes the user comments information (e.g., title, content, time, voting number and the related metadata).

Topic is the collection of multiple post objects and encapsulates a topic title, a topic creating time, the first post, the newest post, the total post number, the total voting number and the related metadata. With those basic functionalities, a user can create and response topic for discussion and relevant posts belong to one topic. When a dataset is uploaded or a script is built by the end user, their corresponding topic will be automictically created. Multiple topics can group as a forum which includes a forum subject, the total topic number, the total post number, the newest post and the related metadata. Forum module provides the following functions through RESTful APIs:

- Query topic by user.
- Query post by user.
- Create and update post.
- Vote a post.
- Query topic list.
- Crete and update topic.
- Query forum.
- Create forum and update the forum description.

3.6 Activity and Auditing

Activity module provides the information of system activities. As shown in Fig. 6, it records the events from Dataset, Script, and Forum modules. It provides internal APIs for these modules to record their activities. The website will get the information from Activity module through RESTful APIs and list the activities. Currently, it supports three functions to list activities:

- List activities of all modules.
- List activities of all modules for a specific dataset.
- List activities of all modules for a specific user.

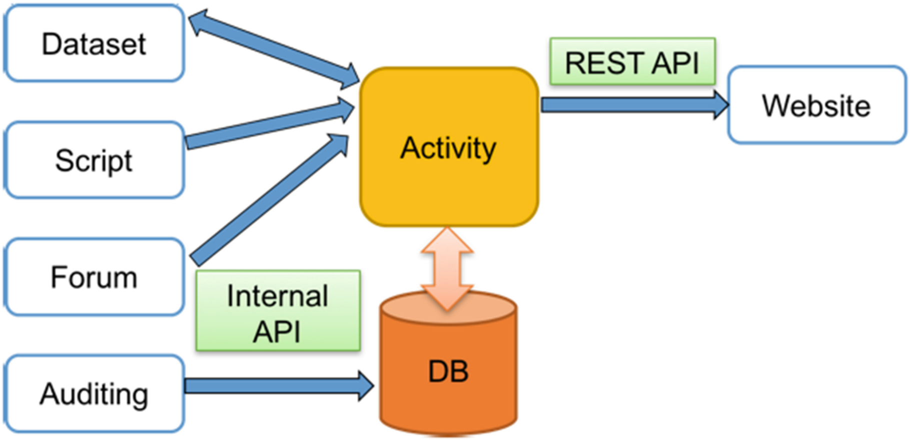


Fig. 6. The relationship between Activity module and other modules.

Since there are various types of events within the system, the amounts of information to record are different. It is necessary to provide a unified data representation to record the activities. The shared columns among all modules are designed as *event_object_n*, where n is from 1 to N (the number N can be decided as needed). Based on this design, since shared column names are fixed, Activity module can also provide a unified interface for other modules. Moreover, it is easy to incorporate NoSQL in the future with little modification. Activity module requires high-performance operations to record activities. In current implementation, the average time of recording one activity is below 9 ms with relational database (i.e. at least 111 activities/second). It is more than good enough for the present requirement. NoSQL database can be integrated easily and flexibly in the future for scalability if activity granularity changes.

Auditing module keeps tracking the resource usages for each user. In Auditing module, there is a daemon periodically asking JupyterHub about the usage. It then parses and records the information in auditing module. Since the parsed information is one kind of activities, it also leverages the data store of Activity module. Currently, only administrators have the privilege to see the information from Auditing module. It shows the computing resources usage over time and can be used for auditing and accounting.

System Evaluation

T-Brain is a collaboration data analytic platform supporting multi-user Jupyter Notebook operations and script running plays a critical role for data scientists to analyze the problem. When opening a script, JypyterHub instantiates Jupyter Notebook container through Docker Swarm. We use Selenium [20] for the T-Brain stress test. Selenium is an automation tool for website. It can simulate user operations with web browsers. It supports various browsers through WebDriver, and supports SDKs for most mainstream languages. In the experiments, we simulate lots of operations from multiple users concurrently. Selenium is used to open, edit, execute, and publish scripts. During the stress test experiments, the resource usage are monitored. The script for analysis is Keras MNIST example program [21]. The following two scenarios are conducting the experiments.

- Operations from multiple users

There are two benchmarks in this experiment: ten and thirty connections operate the system concurrently respectively. During the experiment process (i.e., time units as x-axis), the resource usages (CPU and memory as y-axis) are recorded for JypyterHub (i.e., docker_Jypyterhub) and the Docker container server (i.e., docker_ml_image). From the observations of Figs. 7 and 8, for CPU loading, we can find the results that JypyterHub occupies little resources all the time, while Docker container server occupies the promotional resources during the scripts open. The CPU loading eventually achieves 100%; it slows the analysis but does not introduce any error. For memory usage, it is consuming at most 1,922 MB and 3,717 MB for ten and thirty connections simultaneous operations, respectively. Memory consumption is related with the size of dataset: the larger memory was consuming while the larger the dataset is. It is worth noticing that the memory remains occupied for a while after publishing the script, and it is not available until the Docker container is closed.

- Operations from multiple users with large datasets

When multiple users launch scripts with large datasets (larger than 500 MB) at the same time, it consumes a lot of time waiting for dataset downloading and results in failures due to timeout. As shown in Table 1, in original design of Script module, when ten

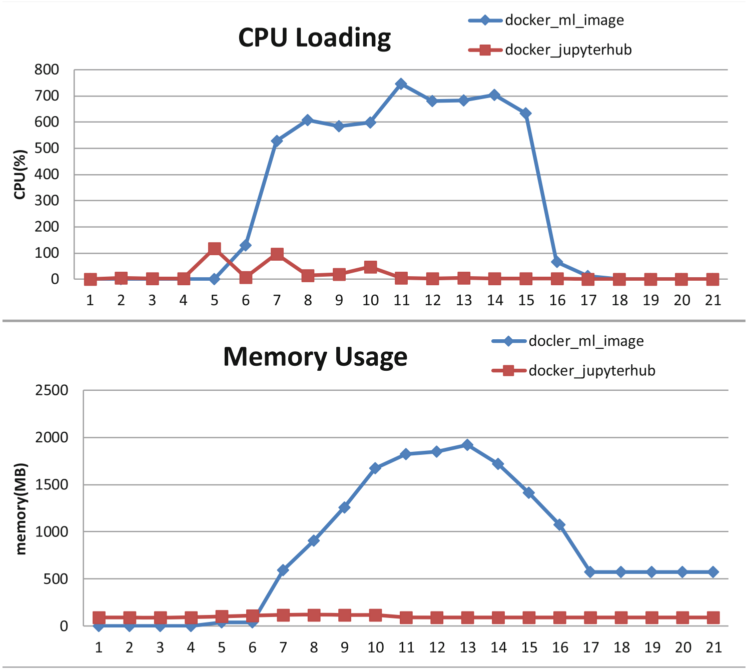


Fig. 7. CPU and memory usage with operations from ten connections

connections open scripts at the same time, only three scripts are opened correctly (i.e., seven scripts are failed to open). The reason is that datasets are downloaded from HDFS, and the bandwidth is not large enough to support dataset finishing downloading within timeout. The default timeout is 10 min. If dataset is not downloaded within timeout, the user will not see the Jupyter Notebook from the website, and it will fail. To overcome this issue, a local cache is introduced. As shown in, Table 2 with the integration of the cache mechanism, the time to read dataset is reduced significantly (average response time improvement 92.36% and request availability improvement 70% by the formula below) and all the scripts can be opened correctly and pass the test.

$$\text{Improvement} = \left(\frac{\textit{time with cache}}{\textit{time without change}} - 1\right) \times 100$$

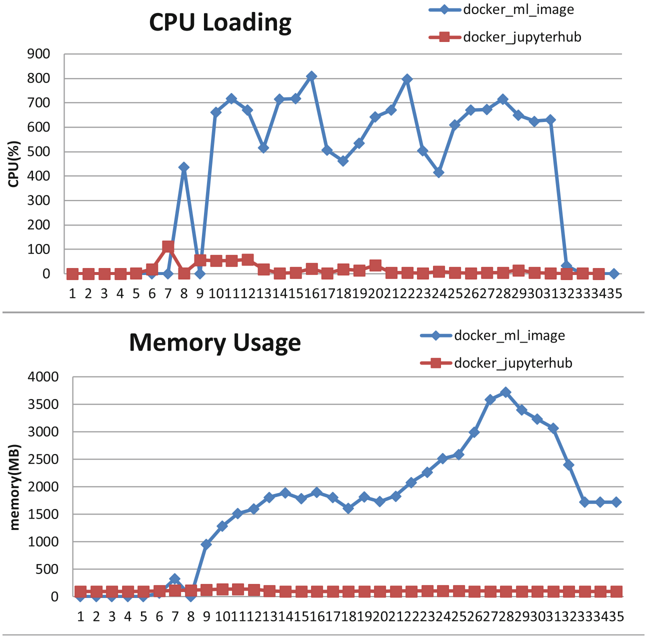


Fig. 8. CPU and memory usage with operations from thirty connections

Table 1. Time for opening scripts in original design

Connections	Time (s)
Test1	161.87
Test2	194.54
Test3	Time out
Test4	Time out
Test5	Time out
Test6	Time out
Test7	Time out
Test8	Time out
Test9	302.07
Test10	Time out
Average time	219.49

Related Works

OpenChorus [22] is an open source collaborative platform for data scientists and developed as a browser based with JavaScript front-end and ruby on rails backend. It provides streamlines (e.g., multiple workspaces within a project) and multi-level secure collaboration (e.g., LDAP [23] and AD [24] based authentication, roles based application

Table 2. Time for opening scripts with local cache design

Connections	Time (s)
Test1	15.01
Test2	15.70
Test3	16.05
Test4	16.07
Test5	15.52
Test6	17.95
Test7	18.05
Test8	19.45
Test9	16.28
Test10	17.32
Average time	16.75

access control and data access control). For analytics tools, it integrates third-party tools (e.g., MADLib [25] and R [26]) and code-design user interface for SQL. For the insight sharing, data scientists can post comments, ask questions and reply answers on any analytics artifacts. It is easy for them to discover and learn from existing insights. OpenChorus opens up potential for data scientist collaboration and improves productivity and performance with big data applications for enterprise companies and start-ups.

Trusted Analytics Platform (TAP) [27] is an open source platform-as-a-Service (PaaS) cloud framework for application developers and data scientists to operate and build the domain-specific applications (e.g., healthcare). TAP includes many well-known and proven open source components (e.g., Kafka [28], Redis [29], Spark [30], HBase [31], MongoDB [32], H2O [33], RStudio [34], iPython [35]) and integrating them as a single platform. It provides good usability for developers and data scientists to collaborate by sharing the analytic environment in private and public clouds.

4 Conclusion

Contribution of our work is that we propose a collaboration platform for data scientists to share the data, write the analytic scripts and discuss the topics related to those problems. In current status, we collect eleven dataset such as spam mail, malware data, honeynet log, Hadoop workload log and some other open data and based on those dataset and improvement local cache design. The platform provide current connections (i.e., average response time improvement 92.36% and request availability improvement 70%) for the data scientist team to analyze the problems. With the platform, there are many education, competition (e.g., HackNTU [36]) and research (e.g., Hadoop configuration optimization [37]) activities can be hold successfully on the collaboration platform.

Acknowledgements. This work was supported in part by Trend Micro, National Center for High-Performance Computing, the Institute for Information Industry under the grant 106-EC-17-D-11-1502 and G367JA1210, Taiwan Information Security Center (TWISC), Academia Sinica, and the Ministry of Science and Technology, Taiwan under the grant 105-2221-E-009-122, and 106-3114-E-009-007.

References

1. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., et al.: Mastering the game of Go with deep neural networks and tree search. Nature **529**, 484–489 (2016)
2. Sadek, I., Elawady, M., Shabayek, A.E.R.: Automatic classification of bright retinal lesions via deep network features. arXiv preprint arXiv:1707.02022 (2017)
3. Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., et al.: An empirical evaluation of deep learning on highway driving. arXiv preprint arXiv:1504.01716 (2015)
4. InnoCentive|Open Innovation & Crowdsourcing Platform. https://www.innocentive.com/
5. Kaggle: Your home for data science. https://www.kaggle.com/. Accessed 1 Dec 2017
6. CrowdANALYTIX: Automating business processes using artificial intelligence
7. CodaLab - Home. https://worksheets.codalab.org/. Accessed 1 Dec 2017
8. Topcoder|Deliver Faster through Crowdsourcing. https://www.topcoder.com/
9. HackerRank|Technical Recruiting|Hiring the Best Engineers. https://www.hackerrank.com/. Accessed 1 Dec 2017
10. Merkel, D.: Docker: lightweight Linux containers for consistent development and deployment. Linux J. **2014**, 2 (2014)
11. Docker Swarm overview|Docker Documentation. https://docs.docker.com/swarm/overview/. Accessed 1 Dec 2017
12. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM **51**, 107–113 (2008)
13. Project Jupyter|Home. http://jupyter.org/. Accessed 1 Dec 2017
14. Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B.E., Bussonnier, M., Frederic, J., et al.: Jupyter Notebooks-a publishing format for reproducible computational workflows. In: ELPUB, pp. 87–90 (2016)
15. GitHub - jupyterhub/jupyterhub: Multi-user server for Jupyter notebooks. https://github.com/jupyterhub/jupyterhub. Accessed 1 Dec 2017
16. Overview of RESTful API Description Languages - Wikipedia. https://en.wikipedia.org/wiki/Overview_of_RESTful_API_Description_Languages. Accessed 1 Dec 2017
17. Deploying JupyterHub for Education. https://developer.rackspace.com/blog/deploying-jupyterhub-for-education/. Accessed 1 Dec 2017
18. Redis. https://redis.io. Accessed 1 Dec 2017
19. Celery: Distributed task queue. http://www.celeryproject.org
20. Selenium - Web Browser Automation. http://www.seleniumhq.org
21. keras-mnist-tutorial/MNIST in Keras.ipynb at master · wxs/keras-mnist-tutorial · GitHub. https://github.com/wxs/keras-mnist-tutorial/blob/master/MNIST%20in%20Keras.ipynb
22. OpenChorus Project: The Dawn of The Data Science Movement|Dell EMC Big Data. http://bigdatablog.emc.com/2012/11/09/openchorus-project-the-dawn-of-the-data-science-movement/
23. Zeilenga, K.: Lightweight directory access protocol (LDAP): technical specification road map (2006)
24. Pierson, N.: Overview of Active Directory Federation Services in Windows Server 2003 R2. Microsoft Corporation, October 2005

25. Hellerstein, J.M., Ré, C., Schoppmann, F., Wang, D.Z., Fratkin, E., Gorajek, A., et al.: The MADlib analytics library: or MAD skills, the SQL. Proc. VLDB Endow. **5**, 1700–1711 (2012)
26. RC Team: R language definition. R Foundation for Statistical Computing, Vienna, Austria (2000)
27. Trusted Analytics. https://github.com/trustedanalytics. Accessed 1 Dec 2017
28. Kafka, A.: A high-throughput, distributed messaging system, vol. 5 (2014). kafka.apache.org
29. Zawodny, J.: Redis: lightweight key/value store that goes the extra mile. Linux Magazine, vol. 79 (2009)
30. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., et al.: Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, p. 2 (2012)
31. Vora, M.N.: Hadoop-HBase for large-scale data. In: 2011 International Conference on Computer Science and Network Technology (ICCSNT), pp. 601–605 (2011)
32. Chodorow, K.: MongoDB: The Definitive Guide: Powerful and Scalable Data Storage. O'Reilly Media, Inc. (2013)
33. H2O.ai. https://www.h2o.ai. Accessed 1 Dec 2017
34. RStudio - Open source and enterprise-ready professional software for R. https://www.rstudio.com
35. Pérez, F., Granger, B.E.: IPython: a system for interactive scientific computing. Comput. Sci. Eng. **9** (2007)
36. HackNTU. https://www.facebook.com/hackNTU/posts/1146642025421019:0. Accessed 3 Dec 2017
37. Yeh, C.-C., Zhou, J., Chang, S.-A., Lin, X.-Y., Sun, Y., Huang, S.-K.: BigExplorer: a configuration recommendation system for big data platform. In: 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI), pp. 228–234 (2016)

Feature Extraction in Security Analytics: Reducing Data Complexity with Apache Spark

Dimitrios Sisiaridis(✉) and Olivier Markowitch

QualSec Group, Departement d' Informatique, Université Libre de Bruxelles,
Brussels, Belgium
{dimitrios.sisiaridis,olivier.markowitch}@ulb.ac.be
https://qualsec.ulb.ac.be/

Abstract. Feature extraction is the first task of pre-processing input logs in order to detect cybersecurity threats and attacks while utilizing machine learning. When it comes to the analysis of heterogeneous data derived from different sources, this task is found to be time-consuming and difficult to be managed efficiently. In this paper we present an approach for handling feature extraction for security analytics of heterogeneous data derived from different network sensors. The approach is implemented in Apache Spark, using its python API, named pyspark.

Keywords: Machine learning · Feature extraction
Security analytics · Apache Spark

1 Introduction

Today, a perimeter-only security model in communication system is insufficient. With the Bring Your Own Device (BYOD) and IoT, data now move beyond the perimeter. For example, threats to the intellectual property and generally to sensitive data of an organisation, are related either to insider attacks, outsider targeted attacks, combined forms of internal and external attacks or attacks performed over a long period. Adversaries can be either criminal organisations, careless employees, compromised employees, leaving employees or state-sponsored cyber espionage.

The augmentation of these cyber security attacks during the last years emerges the need for automated traffic log analysis over a long period of time at every level of the enterprise or organisation information system. Unstructured, semi-structured or structured data in time-series with respect to security-related events from users, services and the underlying network infrastructure usually present a high level of large dimensionality and non-stationarity.

There is a plethora of examples in the literature as well as in open-source or commercial threat detection tools where machine learning algorithms are used to correlate events and to apply predictive analytics in the cybersecurity landscape.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 310–321, 2018.
https://doi.org/10.1007/978-3-319-76451-1_29

Incident correlation refers to the process of comparing different events, often from multiple data sources in order to identify patterns and relationships enabling identification of events belonging to one attack or, indicative of broader malicious activity. It allows us to better understand the nature of an event, to reduce the workload needed to handle incidents, to automate the classification and forwarding of incidents only relevant to a particular consistency and to allow analysis to identify and reduce potential false positives.

Predictive Analytics, using pattern analysis, deals with the prediction of future events based on previously observed historical data, by applying methods such as Machine Learning. For example, a supervised learning method can build a predictive model from training data. This model then is used to make predictions about new observations.

We need to build autonomous systems that could act in response to an attack in an early stage. Intelligent machines could implement algorithms designed to identify patterns and behaviours related to cyber threats in real time and provide an instantaneous response with respect to their reliability, privacy, trust and overall security policy framework.

By utilising *Artificial Intelligence* (AI) techniques leveraged by machine learning and data mining methods, a learning engine would enable the consumption of seemingly unrelated disparate datasets, to discover correlated patterns that result in consistent outcomes with respect to the access behaviour of users, network devices and applications involved in risky abnormal actions, and thus reducing the amount of security noise and false positives. Machine learning algorithms can be used to examine, for example, statistical features or domain and IP reputation.

Along with history- and user-related data, network log data are exploited to identify abnormal behaviour concerning targeted attacks against the underlying network infrastructure as well as attack forms such as man-in-the-middle and DDoS attacks.

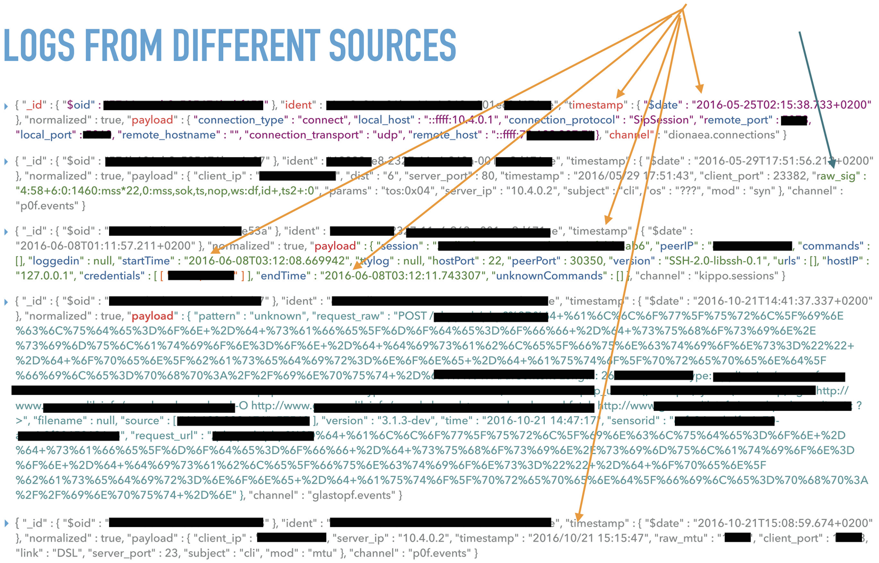
Data acquisition and data mining methods, with respect to different types of attacks such as targeted and indiscriminate attacks, provide a perspective of the threat landscape. Enhanced log data are then analysed for new attack patterns and the outcome, e.g. in the form of behavioural risk scores and historical baseline profiles of normal behaviour, is forwarded to update the learning engine. Any unusual or suspected behaviour can then be identified as an anomaly or an outlier in real or near real-time. In this way, the analysis leverages the integration of credible and actionable threat data to other security devices, in order to protect, guarantee and remediate actual threats, to get insight on how the breach occurred, thus to aid forensic investigations and to prevent future attacks. But, first of all, it is crucial to *extract* and *select* the right data for our analysis, among the plethora of information produced daily by the information system of a company, enterprise or an organisation.

In this paper, we propose an automated approach for *feature extraction* using machine learning methods, as the first stage of a moduled approach for the detection and/or prediction of cybersecurity attacks. For the needs of our experiments we employed the `Spark` framework and more specifically its python API, `pyspark`.

2 Extracting Features from Heterogeneous Data

In our experiments, we examine the case where we have logs of records derived as the result of an integration of logs produced by different network tools and sensors (heterogeneous data from different resources). Each one of them monitors and records a view of the system in the form of records of different attributes and/or of different structure, implying thus an increased level of interoperability problems in a multi-level, multi-dimensional feature space; in the end, each network monitoring tool produces its own schema of attributes.

In such cases, it is typical that the number of attributes is not constant across the records, while the number of complex attributes varies as well. On the other hand, there are attributes, e.g., dates, expressed in several formats, or other attributes referred to the same piece of information by using slightly different attribute names. Most of them are categorical, in a *string* format while the inner datatype varies from nested dictionaries, linked lists or arrays of further complex structure; each one of them may present its own multi-level structure which increases the level of complexity. In such cases, a clear strategy has to be followed for feature extraction. Therefore, we have to deal with *flattening*[1] and *interoperability* solving processes (Fig. 1).



LOGS FROM DIFFERENT SOURCES

```
{ "_id" : { "$oid" : [redacted] }, "ident" : [redacted], "timestamp" : { "$date" : "2016-05-25T02:15:38.733+0200" }, "normalized" : true, "payload" : { "connection_type" : "connect", "local_host" : "::ffff:10.4.0.1", "connection_protocol" : "SipSession", "remote_port" : [redacted], "local_port" : [redacted], "remote_hostname" : "", "connection_transport" : "udp", "remote_host" : [redacted]}, "channel" : "dionaea.connections" }

{ "_id" : { "$oid" : [redacted] }, "ident" : [redacted], "timestamp" : { "$date" : "2016-05-29T17:51:56.213+0200" }, "normalized" : true, "payload" : { "client_ip" : [redacted], "dist" : "6", "server_port" : 80, "timestamp" : "2016/05/29 17:51:43", "client_port" : 23382, "raw_sig" : "4:58+6:0:1460:mss*22,0:mss,sok,ts,nop,ws:df,id+,ts2+:0", "params" : "tos:0x04", "server_ip" : "10.4.0.2", "subject" : "cli", "os" : "???", "mod" : "syn" }, "channel" : "p0f.events" }

{ "_id" : { "$oid" : [redacted] }, "ident" : [redacted], "timestamp" : { "$date" : "2016-06-08T01:11:57.211+0200" }, "normalized" : true, "payload" : { "session" : [redacted], "peerIP" : [redacted], "commands" : [], "loggedin" : null, "startTime" : "2016-06-08T03:12:08.669942", "ttylog" : null, "hostPort" : 22, "peerPort" : 30350, "version" : "SSH-2.0-libssh-0.1", "urls" : [], "hostIP" : "127.0.0.1", "credentials" : [ [ [redacted] ] ], "endTime" : "2016-06-08T03:12:11.743307", "unknownCommands" : [] }, "channel" : "kippo.sessions" }

{ "_id" : { "$oid" : [redacted] }, "ident" : [redacted], "timestamp" : { "$date" : "2016-10-21T14:41:37.337+0200" }, "normalized" : true, "payload" : { "pattern" : "unknown", "request_raw" : "POST /[redacted]4+%61%6C%6C%6F%77%5F%75%72%6C%5F%69%6E
%63%6C%75%64%65%3D%6F%6E+%2D%64+%73%61%66%65%5F%6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%69%6E%2E
%73%69%6D%75%6C%61%74%69%6F%6E%3D%6F%6E+%2D%64+%64%69%73%61%62%6C%65%5F%66%75%6E%63%74%69%6F%6E%73%3D%22%22+
%2D%64+%6F%70%65%6E%5F%62%61%73%65%64%69%72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E%64%5F
%66%69%6C%65%3D%70%68%70%3A%2F%2F%69%6E%70%75%74+%2D%[redacted]http://
www.[redacted]-O http://www.[redacted]http://www[redacted] ?
>", "filename" : null, "source" : [[redacted] ], "version" : "3.1.3-dev", "time" : "2016-10-21 14:47:17", "sensorid" : [redacted]", "request_url" : "[redacted]%64+%61%6C%6C%6F%77%5F%75%72%6C%5F%69%6E%63%6C%75%64%65%3D%6F%6E+%2D
%64+%73%61%66%65%5F%6D%6F%64%65%3D%6F%66%66+%2D%64+%73%75%68%6F%73%69%6E%2E%73%69%6D%75%6C%61%74%69%6F%6E%3D
%6F%6E+%2D%64+%64%69%73%61%62%6C%65%5F%66%75%6E%63%74%69%6F%6E%73%3D%22%22+%2D%64+%6F%70%65%6E%5F
%62%61%73%65%64%69%72%3D%6E%6F%6E%65+%2D%64+%61%75%74%6F%5F%70%72%65%70%65%6E%64%5F%66%69%6C%65%3D%70%68%70%3A
%2F%2F%69%6E%70%75%74+%2D%6E" }, "channel" : "glastopf.events" }

{ "_id" : { "$oid" : [redacted] }, "ident" : [redacted], "timestamp" : { "$date" : "2016-10-21T15:08:59.674+0200" }, "normalized" : true, "payload" : { "client_ip" : [redacted], "server_ip" : "10.4.0.2", "timestamp" : "2016/10/21 15:15:47", "raw_mtu" : [redacted], "client_port" : [redacted], "link" : "DSL", "server_port" : 23, "subject" : "cli", "mod" : "mtu" }, "channel" : "p0f.events" }
```

Fig. 1. Logs from different input sources

[1] The term *flattening* refers to data expressed in 2-D.

3 Global Flattening vs. Local Flattening

The first question to be answered is related to the ability to define an optimal way to handle such complex inputs. Potential solutions may include:

- use the full number of dimensions (i.e. all the available features in each record), defined as *global flattening.*
- decomposing initial logs into distinct baseline structures derived by each sensor/tool, defined as *local flattening.*

3.1 Rationale Behind Our Approach

In order to answer to these questions we should also take into account the rationale behind the next steps. While working with the analysis of heterogeneous data taken from different sources, pre-process procedures, such as *feature extraction*, *feature selection* and *feature transformation*, need to be carefully designed in order not to miss any security-related significant events. These tasks are usually time-consuming producing thus significant delays to the overall time of the data analysis.

That is our main motivation in this work: to reduce the time needed for feature extraction in data exploration analysis by automating the process. In order to achieve it, we utilise the data model abstractions and we keep to a minimum any access to the actual data. The key characteristics of data inputs follow:

- logs derived from different sources
- heterogeneous data
- high-level of complexity
- information is usually hidden in multi-level complex structures

In the next stage, features will be *transformed*, *indexed* and *scaled* to overcome skewness, by following usually a normal distribution under a common metric space, in the form of vectors. As it is about unlabelled data (i.e. lack of any labels or any indication of a suspicious threat/attack), *clustering* techniques will be used to define baseline behavioural profiles and to detect outliers. The latter may correspond to rare, sparse *anomalies*, that can be found by either *first-class* detection of novelties, *n-gram* analysis of nested attributes and *pattern analysis* using Indicators of Compromise (IoCs). Finally, semi-supervised or/and supervised analysis can be further employed by using cluster labels, anomalous clusters, or experts feedback (using *active learning* methods), in order to detect and/or predict threats and attacks in near- and real-time analysis [4].

Outliers in time-series are expected to be found for a i. single network sensor or pen-tester, ii. a subset of those, or iii. by taking into account the complete available set of sensors and network monitoring tools, regarding either:

- time spaces as the contextual attributes
    - date attributes will be decomposed to time windows such as year, month, day of a week, hour and minute, following the approach proposed by [3].

  - statistics will be calculated either for *batch* or *online* mode and then will be stored in HIVE tables, or in temporary views for ad-hoc temporal real-time analysis.
- a single time space (e.g. a specific day)
- a stable window time space (e.g. all days for a specific month)
- a user-defined variable window time space

As our approach serves as an adaptation of the *kill-chain model*[2] *contextual* attributes represent either time-spaces in time-series, as the first level of interest, *single* attributes (e.g. a specific network protocol, or a user or any other atomic indicator), *computed* attributes (e.g. hash values or regular expressions), or even *behavioural* attributes of inner structure (e.g. collections of single and computed attributes in the form of statements or nested records). Then, outliers can be defined for multiple levels of interest for the remain behavioural attributes, by looking into single vector values, or by looking for the covariance and pairwise correlation (e.g. *Pearson* correlation) in a subset of the selected features or the complete set of features [5].

Experiments at the exploratory data stage revealed that the number of single feature attributes in this log were between a range of 7 (the smallest number of attributes of a distinct feature space) up to 99 attributes (corresponding to the total number of the overall available feature space). A fact, that led us to carry on with feature extraction by focusing on flattening multi-nested records separately for each different structure (under a number of 13 different baseline structures).

Thus, the main keys in the proposed approach for feature extraction are:

- extract the right data
  - correlation of the 'right data' can reveal long-term APTs
  - re-usable patterns and trend lines as probabilities are indications of zero-day attacks
  - trend lines may also be used to detect DDoS attacks
- handle interoperability issues
- handle time inconsistencies, date formats, different names for the same piece of information, by extending the NLTK python library [1].

3.2 Global Flattening of Input Data

By following this approach, we achieve a full-view of entities behaviour as each row is represented by the full set of dimensions. On the other hand, the majority of the columns will not have a value or it would be null. A candidate solution

[2] The kill chain model [2] is an intelligence-driven, threat-focused approach to study intrusions from the adversaries perspective. The fundamental element is the indicator which corresponds to any piece of information that can describe a threat or an attack. Indicators can be either atomic such as IP or email addresses, computed such as hash values or regular expressions, or behavioural which are collections of computed and atomic indicators such as statements.

would be to use *sparse vectors* in the next stage of feature transformation, which in turn demands special care for *NaN* and *Null* values (for example, replace them either with the *mean*, the *median*, or with a special value). Most of the data in this stage are *categorical.* We need to convert them into *numerical* in the next stages, as in `Spark`, statistical analytics are available only for data in the form of a *Vector* or of the *DoubleType.*

This solution would perform efficiently for a rather small number of dimensions while it will suffer from the well-known phenomenon of the *curse of dimensionality* for a high number of dimensions, where data appear to be sparse and dissimilar in several ways, which prevents common data modelling strategies from being efficient.

3.3 Local Flattening of Input Data

By following this approach, we identify all the different schemas in input data. First, it is a *bottom-up* analysis by re-synthesizing results to answer to either simple of complex questions. In the same time, we can define hypotheses to the full set of our input data (i.e. *top-down* analysis) thus, it is a complete approach in data analytics, by allowing data to *tell their story*, in a concrete way, following a minimum number of steps. In this way, we are able to:

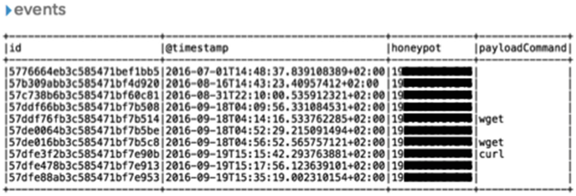
- keep the number of assumptions to a minimum
- look for misconfigurations and data correlations into the abstract dataframes definitions
- keep access to the actual data to a minimum
- provide solutions in interoperability problems, such as:
  - different representations of date attributes
  - namespace inconsistencies (e.g. attributes with names such as prot, protocol, connectionProtocol)
- cope with complex structures of different number of inner levels
- deal with event ordering and time-inconsistencies [6].

4 Feature Extraction in Apache Spark

In `Apache Spark`, data are organised in the form of *dataframes*, which resemble the well-known *relational tables*: there are *columns* (i.e. *attributes* or *features* or *dimensions*) and *rows* (i.e. events recorded, for example, by a network sensor, or a specific device). The list of columns and their corresponded datatypes define the *schema* of a dataframe. In each dataframe, its columns and rows i.e. its schema is unchangeable. An example of a schema follows:

```
DataFrame[id: string, @timestamp: string, honeypot: string,
payloadCommand: string]
```

A sample of recorded events of this dataframe schema is shown in Fig. 2:



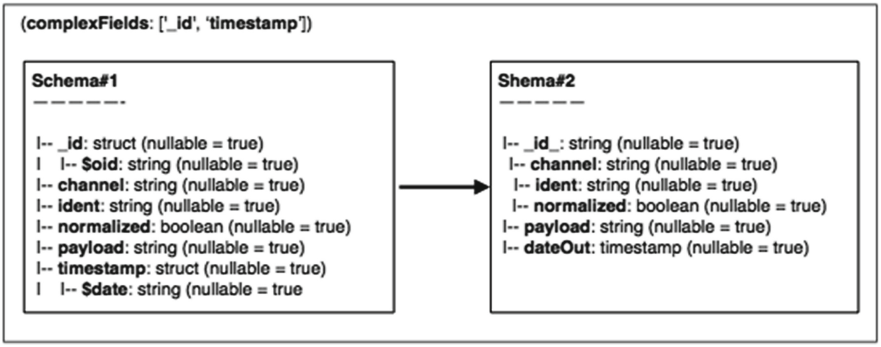
```
+------------------------+-----------------------------------+-------------+--------------+
|id                      |@timestamp                         |honeypot     |payloadCommand|
+------------------------+-----------------------------------+-------------+--------------+
|5776664eb3c585471bef1bb5|2016-07-01T14:48:37.839108389+02:00|19[redacted]|              |
|57b309abb3c585471bf4d920|2016-08-16T14:43:23.40957412+02:00 |19[redacted]|              |
|57c738b6b3c585471bf60c81|2016-08-31T22:10:00.535912321+02:00|19[redacted]|              |
|57ddf66bb3c585471bf7b508|2016-09-18T04:09:56.331084531+02:00|19[redacted]|              |
|57ddf76fb3c585471bf7b514|2016-09-18T04:14:16.533762285+02:00|19[redacted]|wget          |
|57de0064b3c585471bf7b5be|2016-09-18T04:52:29.215091494+02:00|19[redacted]|              |
|57de016bb3c585471bf7b5c8|2016-09-18T04:56:52.565757121+02:00|19[redacted]|wget          |
|57dfe3f2b3c585471bf7e90b|2016-09-19T15:15:42.293763881+02:00|19[redacted]|curl          |
|57dfe478b3c585471bf7e913|2016-09-19T15:17:56.123639101+02:00|19[redacted]|              |
|57dfe88ab3c585471bf7e953|2016-09-19T15:35:19.002310154+02:00|19[redacted]|              |
+------------------------+-----------------------------------+-------------+--------------+
```

Fig. 2. A sample of recorded events

The following steps refer to the case in which logs/datasets are ingested in `.json` format. Our approach examines the data structures on their top-level, focusing on abstract schemas and re-synthesis of previous and new dataframes, in an automatic way. Access to the actual data is only taken place when there is a need to find schemas in dictionaries and only by *retrieving* just one of the records (thus, even if we have a dataframe of million/billions of events, we only examine the schema of the first record/event).

Steps followed for feature extraction

```
A.  load the logfile in a spark dataframe,
    in json format

B.  find and remove all single-valued attributes
    (this steps applies also to the ‘feature selection’ section)

C.  flatten complex structures

    a. find and flatten all columns of complex structure
       (the steps are run recursively, down to the lowest complex
       attribute of the hierarchy of complex attributes)

       1. e.g. struct, nested dictionaries, nested lists,
          arrays, etc.

       2. (i.e. currently those which their value is of RowType

       3. cases:
             a. struct: RowType -> use the leaf column at the last
                level of this struct-column to add it as a new
                column

             b. list: add list elements as new columns

             c. array: split array’s elements and add the relevant
                new columns

             d. dict: steps:
                i.   find all inner schemas for attributes of type
                     Dict, as a list
                ii.  add the schemaType as an index to the
                     original dataframe
                iii. create a list of dataframes, where each one
                     has its own distinct schema
                iv.  flatten all dict attributes, according to their
                     schemas, in each dataframe of the list of
                     dataframes by adding them as new columns

    b. remove all the original columns of complex structure

D. convert all time-fields into timestamps, using distinct
    time fields in the dataframes

E. integrate similar fields in the list of dataframes
```



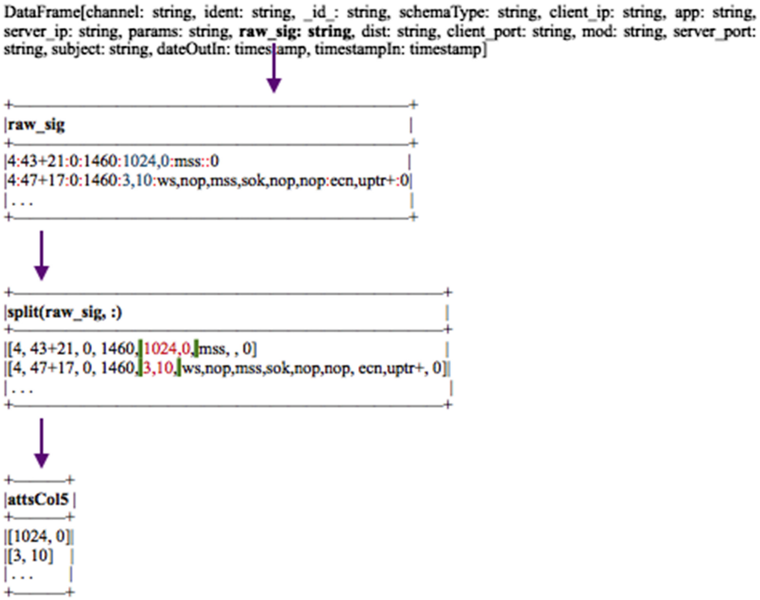
```
(complexFields: ['_id', 'timestamp'])
```

```
Schema#1
------
|-- _id: struct (nullable = true)
|   |-- $oid: string (nullable = true)
|-- channel: string (nullable = true)
|-- ident: string (nullable = true)
|-- normalized: boolean (nullable = true)
|-- payload: string (nullable = true)
|-- timestamp: struct (nullable = true)
|   |-- $date: string (nullable = true
```

```
Shema#2
-----
|-- _id_: string (nullable = true)
 |-- channel: string (nullable = true)
  |-- ident: string (nullable = true)
  |-- normalized: boolean (nullable = true)
|-- payload: string (nullable = true)
|-- dateOut: timestamp (nullable = true)
```

Fig. 3. Transforming complex fields

In this way, we manage to transform the schema of the original dataframe to a number of dataframes, each one corresponding to a schema that refers to a single network sensor or other input data source, as it is illustrated in the following figures (Figs. 3, 4, 5 and 6)



```
DataFrame[channel: string, ident: string, _id_: string, schemaType: string, client_ip: string, app: string,
server_ip: string, params: string, raw_sig: string, dist: string, client_port: string, mod: string, server_port:
string, subject: string, dateOutIn: timestamp, timestampIn: timestamp]

+-----------------------------------------------------+
|raw_sig                                              |
+-----------------------------------------------------+
|4:43+21:0:1460:1024,0:mss::0                         |
|4:47+17:0:1460:3,10:ws,nop,mss,sok,nop,nop:ecn,uptr+:0|
|...                                                  |
+-----------------------------------------------------+

+----------------------------------------------------------+
|split(raw_sig, :)                                         |
+----------------------------------------------------------+
|[4, 43+21, 0, 1460, 1024,0, mss, , 0]                     |
|[4, 47+17, 0, 1460, 3,10, ws,nop,mss,sok,nop,nop, ecn,uptr+, 0]|
|...                                                       |
+----------------------------------------------------------+

+---------+
|attsCol5 |
+---------+
|[1024, 0]|
|[3, 10]  |
|...      |
+---------+
```

Fig. 4. Transforming array fields

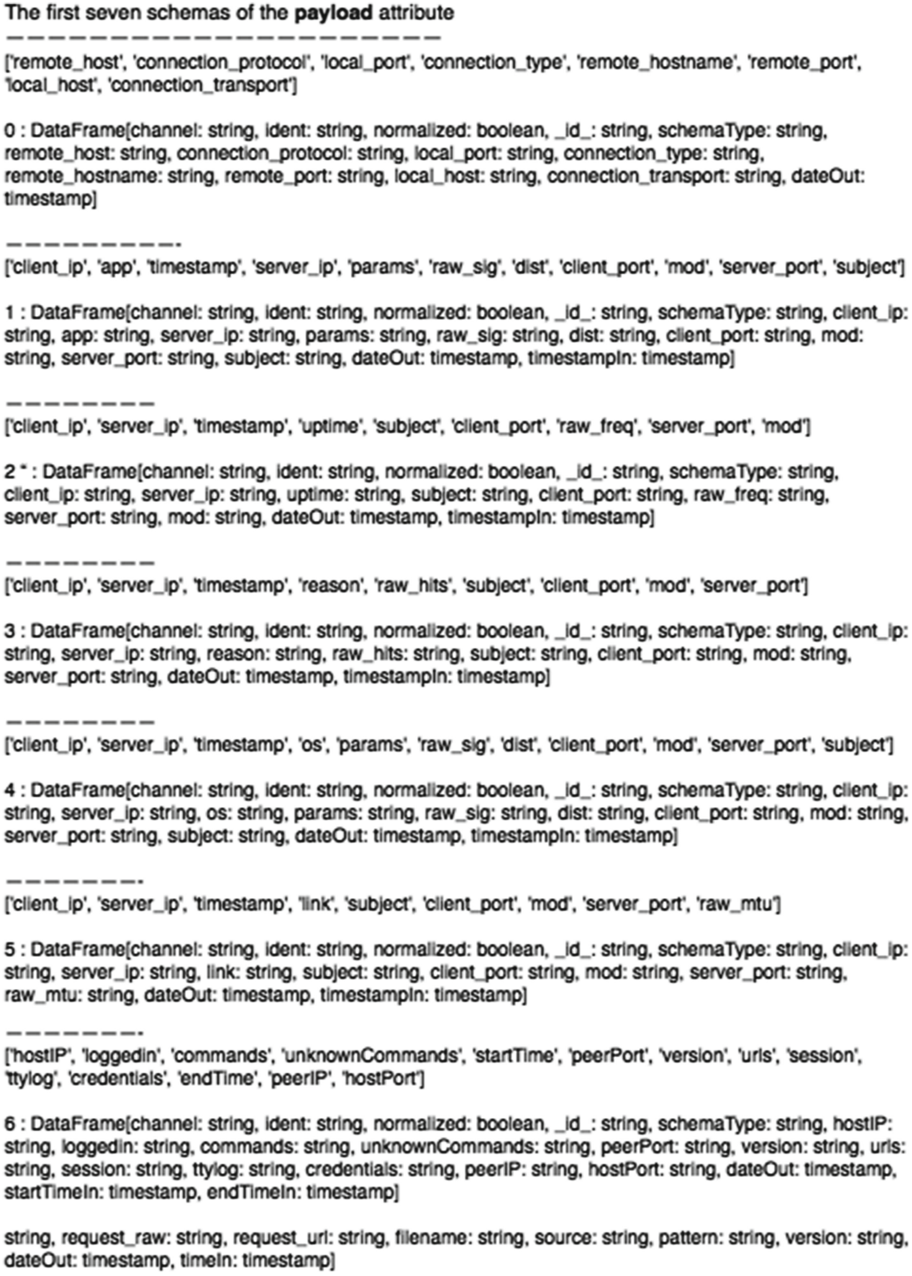
5 Conclusions

We have presented an approach to handle efficiently the task of feature extraction while working with security analytics. It is an automated solution to handle interoperability problems. It is based on a continuous transformation of the abstract definitions of the data inputs, as access to the actual data is limited to a minimum read actions of the first record of a dataframe, and only when it is needed to extract the inner schema of a dictionary-based attribute. The latter is especially important for big data security analytics, while analysing vast amount of heterogeneous data from different sources.

In our experiments we used as input data an integrated log of recorded events produced by a number of different network tools, applied on a telecommunications network.

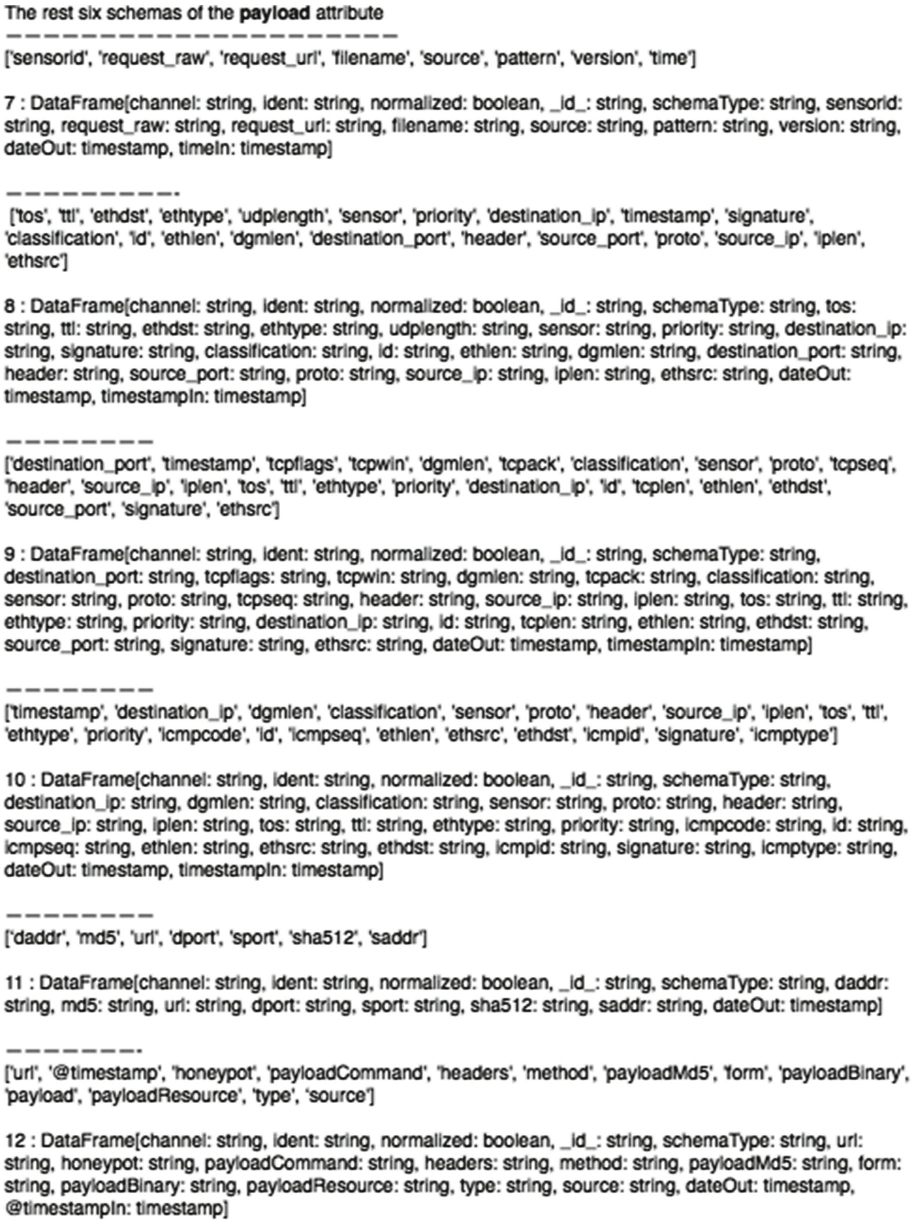
It worths to be mentioned that for this pre-processing analysis stage was used a single server of 2x CPUs, 8 cores/CPU, 64 GB RAM, running an Apache Hadoop installation v2.7 with Apache Spark v2.1.0.

We are currently working into formalizing the approach by utilizing novel structures derived from the theory of categories as it has been presented in [6]. We also plan to proceed with the pre-processing tasks of *feature selection*, *feature cleaning* and *feature transformation* as well as with the actual analysis of heterogeneous logs of unlabelled data in the field of big data security analytics.



```
The first seven schemas of the payload attribute
------------------------

['remote_host', 'connection_protocol', 'local_port', 'connection_type', 'remote_hostname', 'remote_port',
'local_host', 'connection_transport']

0 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string,
remote_host: string, connection_protocol: string, local_port: string, connection_type: string,
remote_hostname: string, remote_port: string, local_host: string, connection_transport: string, dateOut:
timestamp]

----------.
['client_ip', 'app', 'timestamp', 'server_ip', 'params', 'raw_sig', 'dist', 'client_port', 'mod', 'server_port', 'subject']

1 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, client_ip:
string, app: string, server_ip: string, params: string, raw_sig: string, dist: string, client_port: string, mod:
string, server_port: string, subject: string, dateOut: timestamp, timestampIn: timestamp]

--------
['client_ip', 'server_ip', 'timestamp', 'uptime', 'subject', 'client_port', 'raw_freq', 'server_port', 'mod']

2 * : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string,
client_ip: string, server_ip: string, uptime: string, subject: string, client_port: string, raw_freq: string,
server_port: string, mod: string, dateOut: timestamp, timestampIn: timestamp]

--------
['client_ip', 'server_ip', 'timestamp', 'reason', 'raw_hits', 'subject', 'client_port', 'mod', 'server_port']

3 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, client_ip:
string, server_ip: string, reason: string, raw_hits: string, subject: string, client_port: string, mod: string,
server_port: string, dateOut: timestamp, timestampIn: timestamp]

--------
['client_ip', 'server_ip', 'timestamp', 'os', 'params', 'raw_sig', 'dist', 'client_port', 'mod', 'server_port', 'subject']

4 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, client_ip:
string, server_ip: string, os: string, params: string, raw_sig: string, dist: string, client_port: string, mod: string,
server_port: string, subject: string, dateOut: timestamp, timestampIn: timestamp]

-------.
['client_ip', 'server_ip', 'timestamp', 'link', 'subject', 'client_port', 'mod', 'server_port', 'raw_mtu']

5 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, client_ip:
string, server_ip: string, link: string, subject: string, client_port: string, mod: string, server_port: string,
raw_mtu: string, dateOut: timestamp, timestampIn: timestamp]

-------.
['hostIP', 'loggedin', 'commands', 'unknownCommands', 'startTime', 'peerPort', 'version', 'urls', 'session',
'ttylog', 'credentials', 'endTime', 'peerIP', 'hostPort']

6 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, hostIP:
string, loggedin: string, commands: string, unknownCommands: string, peerPort: string, version: string, urls:
string, session: string, ttylog: string, credentials: string, peerIP: string, hostPort: string, dateOut: timestamp,
startTimeIn: timestamp, endTimeIn: timestamp]

string, request_raw: string, request_url: string, filename: string, source: string, pattern: string, version: string,
dateOut: timestamp, timeIn: timestamp]
```

Fig. 5. The different schemas of the payload attribute and the corresponded transformed dataframes (i)



The rest six schemas of the **payload** attribute

```
— — — — — — — — — — — — — — — — — — — — — —
['sensorid', 'request_raw', 'request_url', 'filename', 'source', 'pattern', 'version', 'time']

7 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, sensorid:
string, request_raw: string, request_url: string, filename: string, source: string, pattern: string, version: string,
dateOut: timestamp, timeIn: timestamp]

— — — — — — — — — -
['tos', 'ttl', 'ethdst', 'ethtype', 'udplength', 'sensor', 'priority', 'destination_ip', 'timestamp', 'signature',
'classification', 'id', 'ethlen', 'dgmlen', 'destination_port', 'header', 'source_port', 'proto', 'source_ip', 'iplen',
'ethsrc']

8 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, tos:
string, ttl: string, ethdst: string, ethtype: string, udplength: string, sensor: string, priority: string, destination_ip:
string, signature: string, classification: string, id: string, ethlen: string, dgmlen: string, destination_port: string,
header: string, source_port: string, proto: string, source_ip: string, iplen: string, ethsrc: string, dateOut:
timestamp, timestampIn: timestamp]

— — — — — — — —
['destination_port', 'timestamp', 'tcpflags', 'tcpwin', 'dgmlen', 'tcpack', 'classification', 'sensor', 'proto', 'tcpseq',
'header', 'source_ip', 'iplen', 'tos', 'ttl', 'ethtype', 'priority', 'destination_ip', 'id', 'tcplen', 'ethlen', 'ethdst',
'source_port', 'signature', 'ethsrc']

9 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string,
destination_port: string, tcpflags: string, tcpwin: string, dgmlen: string, tcpack: string, classification: string,
sensor: string, proto: string, tcpseq: string, header: string, source_ip: string, iplen: string, tos: string, ttl: string,
ethtype: string, priority: string, destination_ip: string, id: string, tcplen: string, ethlen: string, ethdst: string,
source_port: string, signature: string, ethsrc: string, dateOut: timestamp, timestampIn: timestamp]

— — — — — — — —
['timestamp', 'destination_ip', 'dgmlen', 'classification', 'sensor', 'proto', 'header', 'source_ip', 'iplen', 'tos', 'ttl',
'ethtype', 'priority', 'icmpcode', 'id', 'icmpseq', 'ethlen', 'ethsrc', 'ethdst', 'icmpid', 'signature', 'icmptype']

10 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string,
destination_ip: string, dgmlen: string, classification: string, sensor: string, proto: string, header: string,
source_ip: string, iplen: string, tos: string, ttl: string, ethtype: string, priority: string, icmpcode: string, id: string,
icmpseq: string, ethlen: string, ethsrc: string, ethdst: string, icmpid: string, signature: string, icmptype: string,
dateOut: timestamp, timestampIn: timestamp]

— — — — — — — —
['daddr', 'md5', 'url', 'dport', 'sport', 'sha512', 'saddr']

11 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, daddr:
string, md5: string, url: string, dport: string, sport: string, sha512: string, saddr: string, dateOut: timestamp]

— — — — — — — -
['url', '@timestamp', 'honeypot', 'payloadCommand', 'headers', 'method', 'payloadMd5', 'form', 'payloadBinary',
'payload', 'payloadResource', 'type', 'source']

12 : DataFrame[channel: string, ident: string, normalized: boolean, _id_: string, schemaType: string, url:
string, honeypot: string, payloadCommand: string, headers: string, method: string, payloadMd5: string, form:
string, payloadBinary: string, payloadResource: string, type: string, source: string, dateOut: timestamp,
@timestampIn: timestamp]
```

Fig. 6. The different schemas of the payload attribute and the corresponded transformed dataframes (ii)

References

1. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O' Reilly Media Inc. (2009)
2. Hutchins, E.M., Cloppert, M.J., Amin, R.M.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. In: Ryan, J. (ed.) Leading Issues in Information Warfare and Security Research, vol. 1, p. 80. Academic Publishing International Ltd., Reading (2011)
3. Kalyan, V., Ignacio, A., Alfredo, C.-I., Vamsi, K., Costas, B., Ke, L.: AI2: Training a big data machine to defend. In: IEEE International Conference on Big Data Security, New York, NY, USA, June 2016
4. Shyu, M.-L., Huang, Z., Luo, H.: Efficient mining and detection of sequential intrusion patterns for network intrusion detection systems. In: Yu, P.S., Tsai, J.J.P. (eds.) Machine Learning in Cyber Trust, pp. 133–154. Springer, Boston (2009)
5. Sisiaridis, D., Carcillo, F., Markowitch, O.: A framework for threat detection in communication systems. In: Proceedings of the 20th Pan-Hellenic Conference on Informatics, pp. 68:1–68:6. ACM (2016)
6. Sisiaridis, D., Kuchta, V., Markowitch, O.: A categorical approach in handling event-ordering in distributed systems. In: Parallel and Distributed Systems (ICPADS), pp. 1145–1150. IEEE (2016)

Storage-Saving Bi-dimensional Privacy-Preserving Data Aggregation in Smart Grids

Chun-I Fan[1], Yi-Fan Tseng[1](✉), Yi-Hui Lin[1], and Fangguo Zhang[2]

[1] Department of Computer Science and Engineering,
National Sun Yat-sen University, Kaohsiung, Taiwan
cifan@mail.cse.nsysu.edu.tw, yftseng1989@gmail.com, blue_6132@hotmail.com
[2] School of Information Science and Technology,
Sun Yat-sen University, Guangzhou, China
isszhfg@mail.sysu.edu.cn

Abstract. Recently, lots of works on power consumption data aggregation have been proposed for the privacy-preservation of users against the operation center in smart grids. This is the *user-based* data aggregation, which accumulates the power consumption data of a group of users for every time unit. On the other hand, the accumulation of a user's data in a group of time units will facilitate the queries on the user's accumulated power usage in these specified time units, which is *time-based* data aggregation. It enables the operation center to perform individual energy consumption statistics and management and offer customized services. If a data aggregation scheme provides both user-based and time-based data aggregation, it is said to be *bi-dimensional*. This manuscript presents the first privacy-preserving bi-dimensional data aggregation scheme, where the storage cost only linearly increases with the number of time units and is independent of the number of users.

1 Introduction

Being regarded as the next-generation energy grid, the developments and researches of smart grids [1,4,7–11,13,14,17,20,21,23,24] have thrived over the world. Nowadays, government of these countries: the U.S., China, Australia, South Korea, and European Community (EC) invested heavily in smart grids [10]. The researches can be roughly classified into three topics [2]: energy management [12,16], information management [3,22] and security [15,18].

In smart gird environments, the power usage is monitored and managed by an operation center in order to adjust the supply and demand curve of power usage and detect threats and failures in real time. In such a system, each user will report the information of her/his power usage every time unit, such as 15 min. The operation center will estimate users' energy consumption of the next time unit with the information, and then distribute energy to users. With the real-time monitoring, smart grid efficiently reduces the energy consumption

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 322–329, 2018.
https://doi.org/10.1007/978-3-319-76451-1_30

compared with traditional architectures. Nevertheless, the frequent monitoring may expose the routines and schedules of users, which causes privacy leakage in smart grids. The data aggregation mechanisms are thus introduced to this environment. The power usage information will be first transmitted to an aggregator. After receiving the data, the aggregator will aggregate every user's data, and send the aggregated data to the operation center. The data received by the operation center have been "accumulated", and thus, they reveal nothing about the private information of each user. This is the *user-based* data aggregation, which provides single-dimensional data aggregation only.

Nevertheless, except the user-based data aggregation, we also require *time-based* data aggregation which allows the operation center to retrieve the accumulated energy consumption of a user for some specified time units. It can support the operation center to do individual energy consumption statistics and management for customized services. For instance, there are several power plants where one gives a much lower price of energy on Mondays but higher on Sundays. The operation center needs to know the energy consumption of each user on Mondays and Sundays in order to provide appropriate or customized discounts to the users.

To achieve both user-based and time-based data aggregation, called *bi-dimensional* data aggregation, a typical approach is to record the power usage of each user in each time unit in the aggregator for the response to any possible query from the operation center. Assume that N is the number of the residential users and M is the total number of time units. Thus, the storage cost of the aggregator will be $O(N \times M)$, which might be enormous.

This manuscript presents the first bi-dimensional data aggregation scheme for smart grids, which requires $O(M)$ storage cost only. Compared to other schemes, the proposed scheme provides lower storage cost and bi-dimensional data queries while achieving privacy preservation simultaneously.

2 Preliminaries

2.1 System Model

In the proposed scheme, we mainly focus on how to send residential users' data to the aggregator privately, without being eavesdropped or intercepted by the operation center. There are three entities in the system model as follows.

- **Operation Center:** The operation center controls the transmission and distribution of electrical energy based on the aggregated data received from the aggregator. In order to achieve privacy preservation, it should be assumed that the operation center does not collude with the aggregator.
- **Aggregator:** The aggregator is mainly responsible for the aggregation of the users' data.
- **Residential Users:** Residential users utilize smart meters to generate electricity usages and report their data to the aggregator.

The proposed scheme includes the following four phases.

- **System Initialization:** In this phase, all entities, including the operation center, the aggregator, and residential users, setup their public and private parameters.
- **Data Generation:** Residential users are equipped with smart meters to record electricity consumption data and compute encrypted data with their signatures, residential area tags, and time stamps. Then, they send the encrypted data to the aggregator.
- **User-Based Data Aggregation:** After receiving encrypted consumption data from users, the aggregator accumulates these data, and sends addressed data to the operation center.
- **Time-Based Data Aggregation:** When the operation center would like to make a query with a set of indexes of time units $\{\overline{1}, \cdots, \overline{M}\}$, which may not be consecutive, it sends the set to the aggregator. After receiving it, the aggregator aggregates the data and responds to the query.

2.2 Security Requirements

Security is a critical issue in smart grids. We consider that the operation center and the aggregator both are honest but curious. However, there exists an adversary A residing in a residential area to eavesdrop the users' reports. In addition, A may also intrude into the database of the operation center or the aggregator to steal the individual user reports. The adversary A could also take some active attacks to alter the data. Therefore, in order to prevent A from learning the users' information, we should meet the security requirements as follows in smart grids.

- **Privacy Preservation:** Adversary A, who intercepts the communications, cannot derive the contents of the data and significant information from the ciphertext and the public key in polynomial time. Furthermore, none of the participated parties, especially the operation center, can catch the detailed consumption data of any user in the region.
- **Authentication and Data Integrity:** All of the reported data should be authenticated, which can ensure that an encrypted report is really sent by a legal residential user and has not been forged or modified during the transmission.

3 The Proposed Scheme

The proposed privacy-preserving bi-dimensional data aggregation scheme for smart grids is presented in this section, where some notations are defined in Table 1. It contains the following phases.

Table 1. The notations

Notation	Meaning
U_i	Residential user i
(PK_i, SK_i)	Public/Secret key pair of residential user i
$d_{i,j}$	U_i's power consumption data of the j-th time unit
M	The total number of time units
N	The total number of residential users
RA	Residential area tag
TS	Timestamp
$\overline{q}$	The maximum number of time units in a query
d	The maximum power consumption in a time unit of a user

3.1 System Initialization

- **Operation Center:** First, the operation center generates a public key (n, g) and the corresponding private key (λ, μ) of the Paillier cryptosystem [19]. Assume that the maximum number of households in a residential area is not greater than a constant N and every user's electricity consumption in a time unit is not greater than d. The operation center chooses a k-superincreasing sequence $\overrightarrow{a} = (a_1, a_2, \cdots, a_{N+2})$ such that $k\sum_{i=1}^{j-1} a_i < a_j, 0 < j \leq N+2$ where $k = \overline{q}d$. It then computes $(g_1, g_2, \cdots, g_{N+2})$, where $g_i = g^{a_i} \mod n^2$, for $i = 1, 2, \cdots, N+2$. After that, it chooses and publishes a digital signature scheme $S = (KeyGen, Sign, Ver)$, the parameters as $pubs = \{(n, g), (g_1, \cdots, g_{N+2})\}$, and keeps $(\lambda, \mu, \overrightarrow{a})$ secretly.
- **Aggregator:** The aggregator chooses an asymmetric encryption scheme $\mathcal{E} = (KeyGen, Enc, Dec)$.
- **User U_i:** Compute $(PK_i, SK_i) \leftarrow S.KeyGen$.

3.2 Data Generation

A user, say U_i, performs the following operations every time unit, e.g., 15 min.

1. Choose a random number $r_{i,j} \in \mathbb{Z}_n^*$.
2. Let $d_{i,j}$ be U_i's power consumption of the j-th time unit. Compute

$$C_{i,j} = (g_i g_{N+2})^{d_{i,j}} r_{i,j}^n \mod n^2.$$

3. Compute $\sigma_{i,j} = S.Sign(C_{i,j} \parallel RA \parallel U_i \parallel TS)$ using SK_i, where RA represents the residential area and TS is the timestamp.
4. Compute $CT_{i,j} = \mathcal{E}.Enc(C_{i,j} \parallel RA \parallel U_i \parallel TS \parallel \sigma_{i,j})$ and send it to the aggregator.

3.3 User-Based Data Aggregation (Data Aggregation for the j-th Time Unit)

After receiving the encrypted data from all users, the aggregator performs as follows.

1. For $i = 1$ to N, compute $(C_{i,j} \parallel RA \parallel U_i \parallel TS \parallel \sigma_{i,j}) \leftarrow \mathcal{E}.Dec(CT_{i,j})$.
2. For $i = 1$ to N, verify the signature $\sigma_{i,j}$ using $S.Ver$.
3. Compute $C_j = \prod_{i=1}^{N} C_{i,j} \mod n^2 =$
$$g^{(\sum_{i=1}^{N} a_i d_{i,j}) + a_{N+2} \sum_{i=1}^{N} d_{i,j}} (\prod_{i=1}^{N} r_{i,j})^n \mod n^2$$
and store C_j.
4. Compute $C'_j = C_j \prod_{i=1}^{N} g_i^{x_{i,j}} \mod n^2$ where $x_{i,j}$, called a blinding factor, is randomly chosen from $[1, k]$ for $i = 1$ to N.
5. Report addressed data C'_j to the operation center.

After receiving C'_j, the operation center retrieves the aggregated power consumption data $m_{*,j}$ of the j-th time unit as follows.

1. Use (λ, μ) to decrypt C'_j and obtain the plaintext
$$m'_{*,j} = (\sum_{i=1}^{N} a_i(d_{i,j} + x_{i,j})) + a_{N+2} \sum_{i=1}^{N} d_{i,j} \mod n.$$
2. Compute
$$m_{*,j} = \frac{m'_{*,j} - (m'_{*,j} \mod a_{N+2})}{a_{N+2}}$$
which equals to $\sum_{i=1}^{N} d_{i,j}$.

Note that $\sum_{i=1}^{N} d_{i,j}$ has been bound with a_{N+2}, not a_{N+1}, so that $\sum_{i=1}^{N} a_i(d_{i,j} + x_{i,j})$ does not perturb $a_{N+2} \sum_{i=1}^{N} d_{i,j}$ even though it overflows into the message space bound with a_{N+1}.

Remark: The aggregator only requires to keep C_j after performing the above protocol, (i.e. *Data Aggregation for the j-th Time Unit*). For M time units, it just needs $O(M)$ storage to keep $\{C_j\}_{1 \leq j \leq M}$, which are enough to provide sufficient data for *Time-Based Data Aggregation.*

3.4 Time-Based Data Aggregation

When the operation center sends a query with a set of indexes of time units $\{\overline{1}, \cdots, \overline{M}\}$ to the aggregator, the aggregator performs as follows.

1. Retrieve $C_{\overline{1}}, C_{\overline{2}}, \cdots, C_{\overline{M}}$ from its storage.
2. Compute $C_{query} = (C_{\overline{1}} \cdot C_{\overline{2}} \cdots C_{\overline{M}}) \mod n^2 =$

$$g^{(\sum_{i=1}^{N} a_i m_{i,*}) + a_{N+2} \sum_{i=1}^{N} m_{i,*}} r^n \mod n^2$$

   where $m_{i,*} = d_{i,\overline{1}} + d_{i,\overline{2}} + \cdots + d_{i,\overline{M}}$ for $i = 1$ to N and r is in $\mathbb{Z}_n^*$.
3. Return C_{query} to the operation center.

After receiving C_{query}, the operation center executes the following steps.

1. Decrypt $g^{-a_{N+2}(m_{*,\overline{1}} + m_{*,\overline{2}} + \cdots + m_{*,\overline{M}})} C_{query} \mod n^2$ which equals to

$$g^{(\sum_{i=1}^{N} a_i m_{i,*})} r^n \mod n^2$$

   and then obtain the plaintext $t_N = \sum_{i=1}^{N} a_i m_{i,*} \mod n$. Note that $m_{*,\bar{i}}$ is the power consumption of the $\bar{i}$-th time unit, which can be obtained by the algorithm shown in Sect. 3.3.
2. For $i = N$ down to 2, compute and output $m_{i,*} = \frac{t_i - (t_i \mod a_i)}{a_i}$, and then compute $t_{i-1} = t_i - a_i m_{i,*}$.
3. Compute and output $m_{1,*} = \frac{t_1}{a_1}$.

The comparison between the proposed scheme and the other existing schemes is summarized in Table 2.

Table 2. Feature comparison

	[1]	[4]	[5][a]	[13]	[20]	Ours
Privacy-preserving against external attackers	Yes	Yes	Yes	Yes	Yes	Yes
Privacy-preserving against internal attackers	No	Yes	Yes	No	Yes	Yes
Assumption on aggregator	Semi-trusted	No Assumption	Semi-trusted	trusted	trusted	Semi-trusted
Authentication and data integrity	Yes	No	Yes	No	Yes	Yes
Bi-dimensional data aggregation	No	No	No	No	No	Yes

[a] Corrections shown in [6] are considered.

4 Conclusion

A novel privacy-preserving data aggregation scheme for smart grids has been presented in the manuscript. The security of Paillier encryption and the unforgeability of the underlying signature can guarantee the security of the proposed scheme. Super-increasing sequences have first been applied to achieve bi-dimensional aggregation while gaining low storage cost. Although adopting super-increasing sequences may cause data expansion, we have exhaustedly utilized the unused message space in Paillier encryption. Compared with the typical approach, the storage cost has decreased tremendously, turning $O(N \times M)$ into $O(M)$. In the future, we will further improve the performance of the scheme. Furthermore, we will attempt to solicit a solution to release the limitation on N or M, which is caused by the data expansion owing to the involving of super-increasing sequences.

References

1. Abdallah, A.R., Shen, X.S.: Lightweight lattice-based homomorphic privacy-preserving aggregation scheme for home area networks. In: 2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1–6. IEEE (2014)
2. Bera, S., Misra, S., Rodrigues, J.J.P.C.: Cloud computing applications for smart grid: a survey. IEEE Trans. Parallel Distrib. Syst. **26**(5), 1477–1494 (2015)
3. Bu, S., Yu, F.R., Liu, P.X.: Dynamic pricing for demand-side management in the smart grid. In: 2011 IEEE Online Conference on Green Communications (GreenCom), pp. 47–51. IEEE (2011)
4. Erkin, Z., Tsudik, G.: Private computation of spatial and temporal power consumption with smart meters. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS 2012. LNCS, vol. 7341, pp. 561–577. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31284-7_33
5. Fan, C.I., Huang, S.Y., Lai, Y.L.: Privacy-enhanced data aggregation scheme against internal attackers in smart grid. IEEE Trans. Ind. Inform. **10**(1), 666–675 (2014)
6. Fan, C.I., Huang, S.Y., Tseng, Y.F.: Corrections to privacy-enhanced data aggregation scheme against internal attackers in smart grid. Technical report (2015). https://doi.org/10.13140/RG.2.1.4006.8649
7. Fouda, M.M., Fadlullah, Z.M., Kato, N., Lu, R., Shen, X.: A lightweight message authentication scheme for smart grid communications. IEEE Trans. Smart Grid **2**(4), 675–685 (2011)
8. Fu, S., Ma, J., Li, H., Jiang, Q.: A robust and privacy-preserving aggregation scheme for secure smart grid communications in digital communities. Secur. Commun. Netw. **9**, 2779–2788 (2015)
9. Galli, S., Scaglione, A., Wang, Z.: For the grid and through the grid: The role of power line communications in the smart grid. Proc. IEEE **99**(6), 998–1027 (2011)
10. Gungor, V.C., Sahin, D., Kocak, T., Ergüt, S., Buccella, C., Cecati, C., Hancke, G.P.: Smart grid technologies: Communication technologies and standards. IEEE Trans. Ind. Inform. **7**(4), 529–539 (2011)
11. Hashmi, M., Hanninen, S., Maki, K.: Survey of smart grid concepts, architectures, and technological demonstrations worldwide. In: 2011 IEEE PES Conference on Innovative Smart Grid Technologies (ISGT Latin America), pp. 1–7. IEEE (2011)
12. Koutitas, G., Tassiulas, L.: A delay based optimization scheme for peak load reduction in the smart grid. In: Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet, p. 7. ACM (2012)

13. Li, F., Luo, B., Liu, P.: Secure information aggregation for smart grids using homomorphic encryption. In: 2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 327–332. IEEE (2010)
14. Li, Q., Cao, G.: Multicast authentication in the smart grid with one-time signature. IEEE Trans. Smart Grid **2**(4), 686–696 (2011)
15. Liu, J., Xiao, Y., Li, S., Liang, W., Chen, C.L.: Cyber security and privacy issues in smart grids. IEEE Commun. Surv. Tutor. **14**(4), 981–997 (2012)
16. Logenthiran, T., Srinivasan, D., Shun, T.Z.: Demand side management in smart grid using heuristic optimization. IEEE Trans. Smart Grid **3**(3), 1244–1252 (2012)
17. Lu, R., Liang, X., Li, X., Lin, X., Shen, X.: Eppa: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. IEEE Trans. Parallel Distrib. Syst. **23**(9), 1621–1631 (2012)
18. Metke, A.R., Ekl, R.L.: Smart grid security technology. In: Innovative Smart Grid Technologies (ISGT), pp. 1–7. IEEE (2010)
19. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
20. Petrlic, R.: A privacy-preserving concept for smart grids. Sicherheit in vernetzten Systemen (2010)
21. Son, H., Kang, T.Y., Kim, H., Roh, J.H.: A secure framework for protecting customer collaboration in intelligent power grids. IEEE Trans. Smart Grid **2**(4), 759–769 (2011)
22. Vytelingum, P., Voice, T.D., Ramchurn, S.D., Rogers, A., Jennings, N.R.: Agent based micro storage management for the smart grid. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, vol. 1, pp. 39-46. International Foundation for Autonomous Agents and Multiagent Systems (2010)
23. Yang, L., Xue, H., Li, F.: Privacy-preserving data sharing in smart grid systems. In: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 878–883. IEEE (2014)
24. Yang, Z., Yu, S., Lou, W., Liu, C.: Privacy-preserving communication and precise reward architecture for v2g networks in smart grid. IEEE Trans. Smart Grid **2**(4), 697–706 (2011)

Verifying the Validity of Public Key Certificates Using Edge Computing

Shogo Kitajima and Masahiro Mambo(✉)

Kanazawa University, Kanazawa, Japan
mambo@ec.t.kanazawa-u.ac.jp

Abstract. Edge computing, which is performed near client, or edge computing together with could computing is expected to provide services with better efficiency than only cloud computing. Meanwhile, most of existing public-key certificate verification methods such as OCSP do not simultaneously achieve efficiency and security with an enough high level. In this paper, we propose a certificate verification method using edge computing and show that our proposed method achieves efficiency with an enough security level by evaluating it through the implementation.

Keywords: PKI · Public key certificate · Edge computing
Web security

1 Introduction

Nowadays, Public Key Infrastructure, PKI, is widely used in internet communication such as in secret communication by SSL/TLS. In the PKI, Certificate Authority, CA, publishes digital certificates describing the correspondence of public key and its owner. The correctness of certificates is guaranteed by trusted CA's digital signature. Once the leakage of secret key or the change of owner's information occurs, certificates are revoked before expiration date. Therefore, certificate revocation needs to be checked even if the expiration date has not yet been passed. Certificate Revocation List, CRL, and Online Certificate Status Protocol, OCSP, are popular method for obtaining revocation status, but these methods have some drawbacks in response speed or privacy preservation. IoT devices are estimated to be widely deployed in the near future. They will communicate with many devices and verify a large number of certificates. Many of such IoT devices have a limited computational power. Therefore, it is important to develop a more efficient certificate verification method suitable for IoT devices. Meanwhile, edge computing [1,2] is technology for improving the communication delay and other features by installing edge server near users. Most of CA servers are not distributed and it often takes much time to obtain revocation status from such distant servers. In this paper, we propose to install revocation status verification server(s) on edge(s). In order to show the efficiency improvement, we evaluate the proposed method by implementation.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 330–336, 2018.
https://doi.org/10.1007/978-3-319-76451-1_31

2 Preliminaries

2.1 Public Key Certificate

Public key certificate is a certificate describing the information of public key's owner. It is used to verify the correctness of a public key for encrypted communication. Today X.509 certificate defined in RFC5280 [3] is a de fact standard. It includes the information of public key, signature algorithm, issuer, expiration date and so on. Client who has received the certificate verifies it. To this end, the client verifies a certificate chain, check the consistency of signatures and obtain revocation status. Some certificates are revoked before expiration date because of the leakage of secret key or the change of owner's information. Therefore, certificate revocation needs to be checked even during the valid period guranteed by the certificate. CRL and OCSP are major methods to verify the revocation status.

2.2 Existing Certificate Revocation Methods

We explain existing certificate revocation methods and summarize their features in Table 1.

CRL: Certificate Revocation List, CRL, [3] is a list of revoked certificates and published by CA. Each certificate issued by the CA specifies URL of the CRL as a download point. Its revocation verification procedure is as follows.

1. Receive a certificate
2. Download CRL specified in the certificate if you do not have it or it is expired.
3. Check whether the serial number of the certificate is included in the CRL or not.

The size of CRL is sometimes very large since it has revocation status of all certificates issued by CA. In terms of the verification of revocation status of a certificate, it contains needless informations and wastes bandwidth. Moreover, it is necessary to cache CRL for every CA and update regularly. In general, CRL is cached but not necessarily kept up-to-date.

OCSP: Online Certificate Status Protocol, OCSP, [4] is an internet protocol to obtain revocation status described in RFC 6960. The implementation is as follows (see also Fig. 2).

1. Client receives a public key certificate.
2. The client creates "OCSP request" and sends it to OCSP responder described in the certificate. OCSP request contains the information of CA and the serial number of the certificate.
3. The OCSP responder checks revocation status of the certificate and returns its result.

Table 1. Comparison of certificate revocation methods

Method	Security	Time	Privacy	Penetration	Independency to the server	Storage	Trust assumption
CRL	✓	✘	✓	✓	✓	✘	CA
OCSP	✓	✛	✘	✓	✓	✓	CA
OCSP stapling	✓	✓	✓	✛	✘	✓	CA
CRLSet	✛	✓	✓	✓	✓	✘	Google
OneCRL	✓	✛	✓	✓	✓	✛	Mozilla
Proposed	✓	✓	✛		✓	✓	Edge

✓: satisfied, ✛: partially satisfied, ✘: not satisfied

while the entire revocation list is downloaded in CRL, only necessary data is obtained through OCSP so that OCSP is not so wasteful as CRL. Also it requires much effort to keep CRL up-to-date, whereas it is easy for clients of OCSP to obtain up-to-date information from intermediate CA. In spite of these advantages, OCSP has drawbacks as follows. Clients which fail to connect to OCSP responder cannot get the status of certificates. Even if the connection to the OCSP responder is successful, there are several issues related to efficiency. OCSP responder needs much bandwidth and it causes burden to CA [6]. If OCSP responder stays in a distant place, it takes much time to obtain status from the responder. Moreover, OCSP request needs to be issued for every certificate in the chain. In addition to the efficiency issues, there is a privacy concern. OCSP responder can obtain client's access information, which should be essentially known only by server.

OCSP Stapling: The TLS extension defined by RFC6066 [5] includes extension of certificate status request and certificate status is sent in TLS handshake. The extention, generally called OCSP stapling, in which ClientHello has certificate status request extension (status_requeststatus_request_v2), checks the certificate status by the following procedure.

1. Server obtains OCSP response from its responder beforehand.
2. Server attaches it to certificate.
3. Client receives the certificate.
4. Client checks the status based on stapled OCSP response.

RFC6066 allows to attach only one response, whereas RFC6961 [6] supports for multiple certificate status, called OCSP Multi-Stapling in general. Since server but not clients requests OCSP and sends status to clients, bandwidth to CA decreases. OCSP response is trustworthy regardless of its source as the OCSP response is signed. However, it does not seem to be enough popular as only 24% (121 out of 500) of HTTPS servers use OCSP Stapling in our investigation in August, 2017.

CRLSet (Implementation on Google Chrome): Google chrome uses CRLSet to ascertain the status.

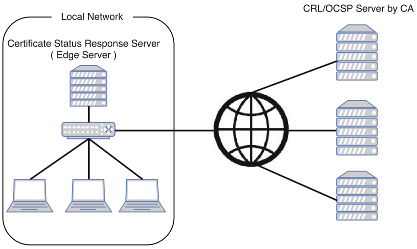


Fig. 1. Network configuration of proposed method

1. Google provides certificate revocation information called CRLSet for Google Chrome.
2. The browser checks the status based on CRLSet.

Google contends "the effectiveness of OCSP is essentially 0 unless the client fails hard (refuses to connect) if it cannot get a live, valid OCSP response. No browser has OCSP set to hard-fail by default". Thus, OCSP is invalid by default in Google Chrome [7]. Even so, since the upper limit size of CRLSet is 250 KB, there is a concern that it may not work well under the revocation of a large number of certificates. This is the reason why the safety is partially satisfied in Table 1.

OneCRL (Implementation on Mozilla Firefox): Mozilla Firefox uses OneCRL to ascertain the status of certificates.

1. Mozilla provides certificate revocation information of intermediate CA called OneCRL for Mozilla Firefox.
2. Client performs one OCSP only for the last certificate at the leaf in a certificate chain, which starts at the leaf and ends at the root.
3. The browser checks the status of the certificate based on the OneCRL and the OCSP.

The differences from CRLSet is that it provides intermediate CA's revocation information. Thus, browsers need to check only the last certificate in the certificate chain by OCSP.

3 Proposed Method

Proposed Protocol

A network of proposed protocol is shown in Fig. 1. The process is shown in Fig. 2 and Fig. 3. We install a revocation status response server near client as shown in Fig. 1. Client checks the status as follows.

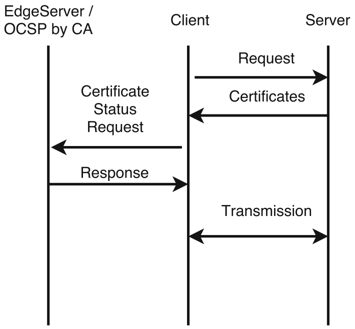


Fig. 2. Process of OCSP and proposed method (Known CA)

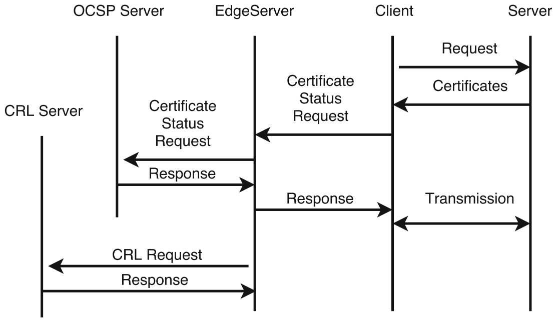


Fig. 3. Process of proposed method (Unknown CA)

1. Client receives public key certificates.
2. Client sends CA information, serial number and URL of CRL and OCSP written on certificate to edge server.
3. If edge server knows the CA, it responses revocation status based on predownloaded CRL (Fig. 2). If it does not know the CA, it processes as follow (Fig. 3).
   (a) Server sends OCSP request to OCSP responder.
   (b) Server replies the status based on OCSP response.
   (c) Server obtains CRL in preparation for the future.

Features

Server needs to be configured to use OCSP Stapling, whereas it does not need in our proposed method. In addition installing near clients makes response time faster. When client keeps CRL himself every client has to request CRL, whereas clients can share CRLs and the burden on the server becomes smaller in the proposed method. Moreover multiple CRLs are managed by edge nodes and client does not need to distinguish different edge nodes and only one edge node is a contact point. Thus client can obtain the status of all certificates by one request.

Table 2. Time for the revocation status check of a certificate [ms]

Percentile	5%	25%	50%	75%	95%
OCSP	11	16	67	214	300
Proposed	0.74	0.88	1.01	1.19	2.1

Table 3. Time for HTTPS communication [ms]

Percentile	5%	25%	50%	75%	95%
OCSP	232	484	922	1621	3650
Proposed	116	233	485	1119	2555

Compared to such methods used in web browsers as CRLSet and OneCRL, client can select trusted resource himself. In terms of storage, such methods of downloading the entire of the revocation list as CRL need more spaces as server using certificate increases. In contrast, such methods of obtaining one or several revocation status as OCSP and the proposed method do not use space and it is suitable for IoT devices. CRL, OCSP and OCSP stapling is lent trustworthiness by CA. Meanwhile, CRLSet, OneCRL and proposed methods is lent it by provider itself because these do not have the signatures by CA. Our proposed method is useful in case that organization installs verification server for its local devices like DNS server.

4 Performance Evaluation

We implemented the client which send HTTPS request using the proposed method and OCSP and the server to response the revocation status in the proposed method by Kotlin. This OCSP client does not use OCSP stapling and uses OCSP for all certificates. We send HTTPS requests to 500 HTTPS servers selected from some user's web browser history.

Table 2 shows time for revocation status check calculated by packet capture. The status check by OCSP induces several OCSP requests as every certificate in the chain needs to be checked. OCSP in Table 2 shows the time for one OCSP request. Table 2 shows the proposed method is 10 100 times faster than OCSP. Moreover, Table 3 shows it contributes to accelerating HTTPS communication.

5 Conclusion

In this paper, we proposed the method using edge computing to verify a public key certificate and we showed it is faster than OCSP. We can construct more efficient system as some neighborhood clients can use the same edge server. Furthermore, edge server can frequently update CRL. By doing so, client of the proposed method can gain higher guarantee on the certificate validity check than client of web browser based implementation.

We used JSON as dataset in the implementation. Thus there is a room for the further speed-up by using more efficient dataset. Moreover, in the implementation, we compared the proposed method to OCSP alone hence we should compare it to other methods. We will also discuss approaches to guarantee the trust of edge server.

References

1. Davis, A., Parikh, J., Weihl, W.: EdgeComputing: extending enterprise applications to the edge of the internet. In: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters, WWW Alternate 2004, pp. 180–187 (2004)
2. Roman, R., Lopez, J., Mambo, M.: Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges. Future Gener. Comput. Syst. **78**, 680–698 (2018)
3. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC5280, May 2008. https://www.ietf.org/rfc/rfc5280.txt
4. Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP, RFC6960, June 2013. https://tools.ietf.org/html/rfc6960
5. Eastlake, D.: 3rd, Transport Layer Security (TLS) Extensions: Extension Definitions, RFC6066, January 2011. https://tools.ietf.org/html/rfc6066
6. Pettersen, Y.: The Transport Layer Security (TLS) Multiple Certificate Status Request Extension, RFC6961, June 2013. https://tools.ietf.org/html/rfc6961
7. Chrome Security FAQ. http://bit.ly/2wbU925

Blockchain Applications in Technology

A Study on Blockchain-Based Circular Economy Credit Rating System

Hsin-Te Wu[1](✉), Yi-Jen Su[2], and Wu-Chih Hu[1]

[1] Department of Computer Science and Information Engineering,
National Penghu University of Science and Technology, No. 300, Liuhe Road,
Magong City 880, Penghu County, Taiwan
wuhsinte@gms.npu.edu.tw

[2] Department of Computer Science and Information Engineering, Shu-Te University,
No. 59, Hengshan Road, Yanchao 824, Kaohsiung County, Taiwan

Abstract. Circular economy is distinct from the linear economy model in the past. Circular economy emphasizes regeneration instead of possession of resource, and proposes using shared resources to create new supply chains and new economies. When practicing circular economy, prior to collaboration, each economic entity must learn of each other's credit rating. This study applies the blockchain technology to establish each economic entity's transaction details, and then employs confidence level algorithms to calculate each entity's credit rating; the method utilizes the concept of decentralization to reduce third-party broker fees, which, aside from decreasing transaction costs, provides effective credit rating of public economic entities.

Keywords: Blockchain · Circular economy · Sharing economy

1 Introduction

Blockchains can alter our future lifestyles. For the Internet of Vehicles, vehicles will be able to utilize blockchains to perform parking and toll payments while also being able to directly pay and download music or multimedia videos. For real estate management, users will be able to use blockchains to lease spare spaces and automatically calculate payments. In the future, blockchains will not only contribute to the Internet of Things or corporations, but also towards food safety, e-voting, intellectual property, and healthcare. Blockchains will completely alter past business models. For financial institutes, consumers will not longer need to conduct procedures such as money transfer personally at the bank during business hours; instead, they will be able to utilize the blockchain technology to perform digital currency transactions. Banks will save up on physical rent fees and labor costs while consumers enjoy secure transaction at any time, any place. The sharing economy can also apply blockchains. Take the ride-sharing service platform Lazooz as an example; users can use the app to search for nearby available vehicles for ride sharing, all the while eliminating the need for an intermediate. We can see from the above why countries around the globe are devoting manpower to researching and developing blockchains because the blockchain technology will bring new economic drive.

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 339–343, 2018.
https://doi.org/10.1007/978-3-319-76451-1_32

Circular economy has risen to become the new generation's economic issue. Circular economy involves leasing, instead of purchasing, idle properties or reusing waste for resource recycling to achieve the goal of sustainable resource management. There are five key concepts to circular economy: (1) redesigning product material: opting for non-disposable and recyclable material for sustainable use of resources; (2) employing ownership-transferring innovative commercial models: changing past linear economy models to substitute buying with leasing; (3) creating higher values through the power of internal circulation: maintaining a product's maximized value by way of circulation, such as utilizing repair, upgrade, reproduction, remarketing to maintain a product's economic value; (4) turning waste into resource: recycling discarded goods and returning them to another product's circulation; (5) establishing industrial symbiosis: bringing different industries to the same region so they may exchange resources, share infrastructures, reduce disposable waste, and lower production costs. This study proposes a trust mechanism for the sharing economy. Many industries currently hold idle machinery or idle space; through a sharing economy, they can increase corporate profits by substituting buying with leasing. However, mutual trust is required in realizing sharing economy; moreover, in order to enable enterprises to share resources at any time and place, trust between enterprises must be transparent and non-modifiable; hence, this study proposes a trust verification mechanism based on blockchain technology that can help realize sharing economy.

This study employs blockchains in establishing the buyer's and seller's credit rating so that the two parties may utilize credit ratings to select a trustworthy business partner. This study utilizes public-key cryptography from blockchains to ensure a transaction's non-repudiation; each party to the transaction can acquire their counterparty's credit ratings by means of verification. Our proposed rating system holds different calculation methods for the buyer and the seller: the seller's involves calculation of corporate capital, transaction amount, and completed transaction progress while the buyer's involves calculation of corporate capital, transaction amount, and payment status, and as for the rating part, the two parties provide ratings for each other. The proposed credit rating system aims to create a better transaction environment for the sharing economy and enable the buyer and seller to choose better business counterparties through transparent credit ratings.

2 Related Work

Ever since Bitcoin's development in 2009, many businesses have dedicated themselves towards Bitcoin. However, because Bitcoin has undergone serious fluctuations in stock prices, some Bitcoin companies experienced bubble burst. Reference [1] offers an analysis model for Bitcoin price prediction to help investors in their Bitcoin investments. Reference [2] offers simulation of the Bitcoin system model; it also simplifies blockchains and avoids double spending risks. Reference [2] simulates blockchains' execution efficiency, and its experiment showed promising results. Reference [3] discusses Bitcoin mining efficacy in Bitcoin software and hardware. Bitcoin has launched digital currency in many countries, and has even established Bitcoin e-payment systems in convenient stores. Reference [4] analyzes the advantages of digital currency and takes a look at Bitcoin's usage of zk-SNARK transaction authentication system. Bitcoin has drastically matured since its introduction in 2009, which has also served as a motivation for blockchain development.

Reference [5] mentions using data envelopment analysis to analyze and verify the effects of circular economy. Reference [6] elaborates on the definition of sharing economy and emergent collectives, and provides case studies of sharing economy. Reference [7] proposes a sharing economy model for public enterprises and private companies that allows resources to be adequately allocated by means of accords and thus increases the scale of sharing economy. Meanwhile, Reference [8] posits the required conditions for a complete transaction and, after analyzing them, concludes that trust is the foremost condition among all. Reference [9] utilizes location sharing to achieve privacy protection and trust, and is mainly applied in social networking sites.

3 The Proposed Scheme

This study proposes a blockchain-based sharing economy credit rating system. The blockchain technology part focuses on utilizing public-key cryptography to verify transaction details. We assume that both the seller and the buyer possess a citizen digital certificate ($\mathcal{C}_{\mathbb{ID}_u}$), in which IDu stands for the user's true ID, the user's public key is $\mathcal{PK}_{\mathbb{ID}_u}$, and private key is $\mathcal{PR}_{\mathbb{ID}_u}$. Each time a seller and buyer engage in a successful transaction, the following information ensues:

$$\mathcal{M}_{\mathbb{ID}_{S_1}} = \mathbb{ID}_{S_1} || \mathbb{ID}_{B_1} || \mathbb{ID}_{B_2} || \ldots || \mathcal{TM}_{\mathbb{ID}_{S_1}} || \mathcal{TA}_{\mathbb{ID}_{B_1}} || \ldots || \mathcal{TR}_{\mathbb{ID}_{B_1}} ||.. \tag{1}$$

$$\mathcal{M}_{\mathbb{ID}_{B_1}} = \mathbb{ID}_{S_1} || \mathbb{ID}_{B_1} || \mathcal{TM}_{\mathbb{ID}_{S_1}} || \mathcal{TA}_{\mathbb{ID}_{B_1}} || \mathcal{TR}_{\mathbb{ID}_{B_1}} \tag{2}$$

Formula (1) concerns the transaction status between the seller and the buyer; $\mathcal{M}_{S_1}$ stands for the information of Seller $\mathcal{S}_1$; stands for the true identity of Seller $\mathcal{S}_1$; $\mathcal{TM}_{S_1}$ is the total transaction amount between Seller $\mathcal{S}_1$ and the buyer $\mathcal{B}_1$; $\mathcal{TA}_{B_1}$ is the transaction amount between $\mathcal{B}_1$ and $\mathcal{S}_1$; $\mathcal{TR}_{B_1}$ is the transaction result between $\mathcal{B}_1$ and $\mathcal{S}_1$, which includes message transaction status, payment time, and completion time. The seller and buyer upload the transaction result to each node. The algorithm is as follows:

$$\mathcal{PR}_{\mathbb{ID}_{S_1}}\left(\mathcal{M}_{\mathbb{ID}_{S_1}}\right) || H\left(\mathcal{M}_{\mathbb{ID}_{S_1}}\right) || \mathcal{C}_{\mathbb{ID}_{S_1}} \tag{3}$$

$$\mathcal{PR}_{\mathbb{ID}_{B_1}}\left(\mathcal{M}_{\mathbb{ID}_{B_1}}\right) || H\left(\mathcal{M}_{\mathbb{ID}_{B_1}}\right) || \mathcal{C}_{\mathbb{ID}_{B_1}} \tag{4}$$

In Formula (3), $\mathcal{S}_1$ uses the private key to encrypt the information and then transmits it along with the certificate to the network; when other nodes receive the message, they use the certificate's public key to decipher the true ID and verify the information's authenticity. In Formula (4), $\mathcal{B}_1$ uses the private key to encrypt the information and then send it along with the certificate to the network; each node then verifies the information's authenticity.

When another user, $\mathcal{B}_2$, wishes to join $\mathcal{S}_1$ in a sharing economy transaction, $\mathcal{B}_2$ can utilize $\mathcal{S}_1$'s true ID to inquire transaction statuses and calculate each transaction amount's credit rating; the calculation is as follows:

$$\mathcal{R}_{\mathbb{ID}_{S_1},i} = \left\{ \begin{array}{l} \left(\mathcal{T}_{\mathbb{ID}_{S_1},i} * 0.3\right) + \left(F_{\mathbb{ID}_{S_1},i} * 0.7\right), TR_{\mathbb{ID}_{S_1},i} = 1 \\ 0, TR_{\mathbb{ID}_{S_1},i} = 0 \end{array} \right\} \tag{5}$$

$$\mathcal{T}_{\mathbb{ID}_{S_1},i} = \left\{ \begin{array}{l} 1, \dfrac{\mathcal{TM}_{\mathbb{ID}_{S_1},i}}{\mathcal{CA}_{\mathbb{ID}_{S_1}}} > 1 \\ \dfrac{\mathcal{TM}_{\mathbb{ID}_{S_1}}}{\mathcal{CA}_{\mathbb{ID}_{S_1}}}, \dfrac{\mathcal{TM}_{\mathbb{ID}_{S_1}}}{\mathcal{CA}_{\mathbb{ID}_{S_1}}} <= 1 \end{array} \right\} \tag{6}$$

$$\mathcal{CR}_{\mathbb{ID}_{S_1}} = \sum_{1}^{n} \mathcal{R}_{\mathbb{ID}_{S_1},i}/n \tag{7}$$

Formula (5) calculates single transaction ratings of Seller $\mathcal{S}_1$ and focuses on the completed transaction of $TR_{\mathbb{ID}_{S_1},i}$. Hence, if $TR_{\mathbb{ID}_{S_1},i}$ is 0, then the transaction rating is 0; if $TR_{\mathbb{ID}_{S_1},i}$ is 1, then the transaction completion time $\mathcal{F}_{\mathbb{ID}_{S_1},i}$ is used to calculate whether it falls within the allotted transaction time; adding the above to the calculation of whether transaction amount $\mathcal{TM}_{\mathbb{ID}_{S_1},i}$ falls within the capital $\mathcal{CA}_{\mathbb{ID}_{S_1}}$, we can obtain the seller's rating $\mathcal{R}_{\mathbb{ID}_{S_1},i}$ for a single transaction. Formula (7) calculates the seller's overall rating; Seller $\mathcal{S}_1$ can also use Formulas (5)–(7) to calculate Buyer $\mathcal{B}_2$'s rating, so that the two parties may understand each other's past number of transactions and their ratings to facilitate their selection of optimal transaction counterparties.

4 Conclusion

This study has here proposed a blockchain-based circular economy credit rating system. In the future, circular economy will be a key point in the government's promotions; however, circular economy's sharing mechanism requires the addition of a credit rating mechanism. In the past, enterprises had to rely on credit checking to obtain information on the other party's credit status, yet this can be time-consuming and cost-increasing for a transaction. This study's proposed credit rating system uses blockchains to conduct network verification; the system's decentralization feature reduces costs of credit investigation while also enabling two parties to a transaction to conduct inquiries at any time and place and facilitate their transaction.

References

1. Li, X., Wang, C.A.: The technology and economic determinants of cryptocurrency exchange rates: the case of Bitcoin. Decis. Support Syst. **95**, 49–60 (2017)
2. Tschorsch, F., Scheuermann, B.: Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. **18**(3), 2084–2123 (2016)
3. Vranken, H.: Sustainability of bitcoin and blockchains. Sustain. Gov. Transform. **28**, 1–9 (2017)
4. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Santiago, Chile, pp. 487–499 (1994)
5. Wang, Y., Liang, B.: Efficiency evaluation of city circular economy based on the super-efficient mixed DEA cluster model. In: Proceedings of the 2010 International Conference on Management and Service Science (2010)
6. Petrie, C.: Emergent Collectives Redux: The Sharing Economy. IEEE Computer Society **20**(4), 84–86 (2016)
7. García, J.M., Fernández, P., Ruiz-Cortés, A., Dustdar, S., Toro, M.: Edge and cloud pricing for the sharing economy. IEEE Internet Comput. **21**(2), 78–84 (2017)
8. Viardot, E.: Trust and standardization in the adoption of innovation. IEEE Commun. Stand. Mag. **1**(1), 31–35 (2017)
9. Schlegel, R., Chow, C.-Y., Huang, Q., Wong, D.S.: Privacy-preserving location sharing services for social networks. IEEE Trans. Serv. Comput. **10**(5), 811–825 (2017)

Using Blockchain to Support Data and Service Management in IoV/IoT

Obaro Odiete[1](✉), Richard K. Lomotey[2], and Ralph Deters[1]

[1] University of Saskatchewan, Saskatoon, Canada
obaro.odiete@usask.ca, deters@cs.usask.ca
[2] Pennsylvania State University, Monaca, PA, USA
rkl5137@psu.edu

Abstract. Two required features of a data monetization platform are query and retrieval of the metadata of the resources to be monetized. Centralized platforms rely on the maturity of traditional NoSQL database systems to support these features. These databases for example MongoDB allows for very efficient query and retrieval of data it stores. However, centralized platforms come with a bag of security and privacy concerns, making them not the ideal approach for a data monetization platform. On the other hand, most existing decentralized platforms are only partially decentralized. In this research, we developed Cowry, a platform for publishing of metadata describing available resources (data or services), discovery of published resources including fast search and filtering. Our main contribution is a fully decentralized architecture that combines blockchain and traditional distributed database to gain additional features such as efficient query and retrieval of metadata stored on the blockchain.

Keywords: IoV · IoT · Data monetization · Blockchain · MultiChain

1 Introduction

The Internet of Things (IoT) [2,24] envisions Internet-connected everyday objects such as cookers, microwaves, fridges communicating and exchanging data. The Internet of Vehicles (IoV) [1] is a type of IoT in the traffic vertical. The primary drivers for IoV are:

- **Consumers:** Market research shows that connecting devices, services and data is an increasingly important element in the purchasing decision. We assume that the connectivity will become even more important when consumer move from purchasing a vehicle to subscribing/renting/sharing.
- **Manufacturers:** With the inevitable move towards electric powered vehicles (at least in urban environments), it becomes difficult for manufacturers to differentiate their products. An electric car is significantly less complex and would be similar to the competitors' model. This, in turn, forces the manufacturers to differentiate via software functionality. Selling software features,

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 344–362, 2018.
https://doi.org/10.1007/978-3-319-76451-1_33

offering subscriptions to services, and finally monetizing the user data are distinct ways to open up the much-needed revenue streams.
- **Cities:** Given the rising number of vehicles on the streets in urban environments, it is inevitable that cities must have more control. Only by communicating with the vehicles it becomes possible to handle increased traffic volumes.
- **Society:** Finally, societies must be able to control and monitor traffic movements to ensure that the right incentive/disincentive mechanisms are deployed to obtain an acceptable balance between unmanaged individual and managed public traffic.

Those four drivers will transform vehicles into platforms that offer services, collect data, and are designed to be part of an evolving digital ecosystem. However, for the data to be useful, thousands (even millions) of data points need to be aggregated from several sources or data owners. For example, for the analysis of traffic patterns, several participants need to provide their car sensor data. There are essentially three main participants involved in this data flow or exchange:

- **Data Owners:** The users (individuals or organizations) that [owns the vehicles that] generates the data.
- **Data Aggregators:** The users that aggregates and process the data for consumption either as is or as a service. For example, organizations interested in collecting data to build services on top, car manufacturers collecting data from sensors installed on their product.
- **Data Consumers:** The users that consumes the processed data. This includes users of services built on top of aggregated data, and users of individual data points.

Nevertheless, most of the financial incentives for generating this data goes to centralized organizations. Individuals have little value in terms of monetary benefits from providing their data, and most times do not have control of what is collected about them or done with the data. A case in point is Facebook conducting experiments on user's data without permission [12]. This is besides the other privacy and security concerns as the data is collected into huge data silos making them targets for hackers [11] and government surveillance [15]. On the bright side, attention is now being given to this area. For example, Tim Berners-Lee, known as the creator of the world wide web envisions a better web where users have more control over their data – where it is stored and how it's accessed [9]. Also, mobile applications such as Google Opinion Reward on Google Play and Survey.com allows users to profit from answering questionnaire surveys selectively presented to them [25]. This model rewards data owners for providing their data.

Blockchain has recently attracted a lot of attention across many different sectors for its ability to decentralize systems and remove the need for a middleman. Blockchain is a decentralized ledger replicated across participating nodes in a peer-to-peer (P2P) network. It is secured by strong cryptographic algorithms ensuring that no one node has full control over what is written into the ledger.

Most existing work for data monetization using blockchain infrastructure combines traditional database technology and blockchain in ways that compromises the decentralization. For example, using Bitcoin's blockchain for the payment infrastructure while still storing the sensor or resources metadata on a centralized database to leverage the robust query and retrieval mechanism it supports. In this paper, we report on a decentralized architecture for data and service monetization based on blockchain technology. We developed Cowry, a platform for publishing of metadata describing available resources (data or services), discovery of published resources including fast search and filtering. Our main contribution is a fully decentralized architecture that combines blockchain and traditional distributed database to gain additional features such as efficient query and retrieval of metadata stored on the blockchain.

2 Blockchain

Blockchain is not a new technology but an innovative marriage of ideas from well-established fields such as public key cryptography [20], distributed consensus [10] and peer-to-peer networking [19]. Blockchain is essentially a chain of blocks all of which are maintained on participating nodes, which do not fully trust each other, in a P2P network. Each block contains an ordered list of events (called transactions) mutually agreed upon by all nodes in the network. The "chain" results from each block referencing the cryptographic hash of the previous block; the first block, called the genesis block does not reference any block. A distributed consensus algorithm ensures the nodes agree on the block's content through a process called Mining.

2.1 Blockchain Ledger

Blockchain ledger is a distributed data structure comprising of "blocks" linked together to form a chain. It was introduced with Bitcoin to solve the fundamental problem of distributed digital currency - double spending [3] which is previously trivially solved using a central authority. Blockchain removes the need for a central trusted entity like a bank since all participants would have the full record of all transactions - according to Nakamoto [22], "the only way to confirm the absence of a transaction is to be aware of all transactions".

Blocks: The structure of blockchain can be described as similar to a linked list with the nodes in the linked list representing the blocks. Each block has a header and a body. The block is identified by the cryptographic hash of its header. The header contains a version number to indicate the rules used to verify the validity of the block, the hash of the previous block header (this is what "chains" the blocks together), the root of the Merkle tree [18], which is a hash of all the transactions in the block, the current Unix timestamp, and a nonce. The body of the block contains the number of transactions and a list of the transactions.

Transactions: Transactions are instructions that assigns ownership right for an amount of digital resource from the current owner signing the transaction to the new owner specified in the transaction. The transaction is signed by the private key of the sender and can be verified by the receiver using the sender's public key. The format of a transaction depends on the blockchain network, however in general it includes the sender and recipient address, data payload and the amount. After transactions are created by participating nodes, they are propagated through the network in a P2P manner and verified by each node before propagating further. Transactions can also be used to store arbitrary data on the blockchain. This feature has been exploited in Bitcoin for different use-cases such as notarizing the existence of a document or as a permanent decentralized data store.

2.2 Blockchain Network

The blockchain network is a decentralized P2P network where each user or participants interacts with the network via their node. To join the network, a compatible blockchain client is installed on the node. The network is decentralized because there is no central server and continues to function even if some nodes leave the network. Data from each node are validated by the receiving node and then forwarded to other nodes it is connected to thus it is possible for a node to receive multiple copies of the same data.

2.3 MultiChain

MultiChain [6] is a platform for creating and deploying private blockchains. The motivation for the development of MultiChain was to solve some of the problems identified with the use of Bitcoin, from which it was derived, for institutional financial transactions. As a platform for private blockchain, it introduces features that ensures only permitted nodes can participate in the network activities including connecting, mining, and sending or receiving transactions. These permissions are configurable during the setup of the network and includes Boolean parameters such as anyone-can-connect, anyone-can-mine etc. The MultiChain permissions documentation [7] provides more details on these and other supported permissions. MultiChain differs from the Bitcoin in several ways including different consensus mechanism – it uses a scheme called Mining Diversity – direct support for third party assets and support for database-like feature via MultiChain Streams [13].

Mining Diversity: MultiChain uses a round-robin scheme to determine which permitted miner can append blocks to the chain. The idea is to limit the number of blocks that a single miner can append within a given window using a network parameter called mining diversity which can be set to a value between 0 and 1 (inclusive) during the blockchain setup. Nodes only attempt to mine if they have not mined any one of the $spacing - 1$ previous blocks where $spacing$ is calculated by $\#permitted\_miners * mining\, diversity$ else their blocks will be invalid.

Assets: Although a fork of Bitcoin core, MultiChain goes a step further by allowing arbitrary third party tokenized assets (virtual tokens representing real world assets) to be created and exchanged on the Blockchain. It enforces the same or higher level of cryptographic security for the transfer of these assets. One use of this feature is to create application-specific cryptocurrencies different from the native one provided by the platform.

Streams: Another feature MultiChain support is called Streams. A MultiChain Stream [14] is an append-only collection of items, implemented underneath as blockchain transactions. This abstraction allows the blockchain to be used for data retrieval and archival. Each item in a stream has 4 fields namely publisher(s), key, data, and timestamp. The key field allows data to be stored and retrieved like in a key-value database, the timestamp enables stored data to be retrieved in time order and the publisher field categorizes the items by their authors for retrieval. By this implementation, a Stream allows three types of databases on top the blockchain:

- Key-value database or document store
- Time series database
- Identity driven database

Oracle: Bitcoin and MultiChain supports two runtime parameters: blocknotify and walletnotify, that allows external scripts to be run in response to some transaction activity on the blockchain [8]. This external script can be seen as an Oracle. An oracle is anything that is used to connect the blockchain to the off-chain world. In our use-case, the oracle is a script that is triggered when certain transaction activities occur on the blockchain. However, the use of oracle goes beyond this use case of responding to a transaction. It is also very important in retrieving input from the off-chain world.

2.4 Blockchain Characteristics

One of the primary feature of the blockchain is its ability to allow direct interaction between non-trusting parties, removing the need for a trusted authority. This feature provides many additional benefits including lower costs (no need for expensive central servers and backups) and redundancy. Other characteristics of blockchain as discussed in the literature [5,33] are:

- **Persistency:** Once the transaction has been recorded on the blockchain, it cannot be easily modified or falsified.

- **Anonymity:** The real identity of the user is not revealed during interactions on the blockchain as only a generated address is used for the transactions. Although, this is not a guarantee of perfect anonymity as a careful analyst can make connections between addresses and may be able to infer the user's real identity from such connections [17,27].
- **Fault Tolerance:** The blockchain ledger is replicated across all nodes, hence providing redundancy even if any node fails or leaves the network.
- **Transparency:** Every participant of the network sees the same state of the transactions recorded on the blockchain.
- **Traceability:** Traceability is an extension of the persistency and transparency characteristics. Every transaction can be audited back till its very beginning.

2.5 Blockchain Challenges

Despite the potentials of blockchain technology, it has some problems usually associated with the public blockchain. Two main challenges found in the literature are:

- **Scalability and Performance:** This is considered one of the main criticism for public blockchains in the literature especially by developers of private blockchain platforms. Bitcoin for example can only process an average of seven transactions per seconds. Other public blockchain platform have similar performance limits especially when compared to traditional databases. Also, blockchains does not scale well since each additional node still replicates and processes the same transactions.
- **Data and User Privacy:** Since all the data on the blockchain are visible to all the participants, it does not support data privacy by default. This challenge is a major drawback especially in finance and legal use cases where data privacy is very crucial. Also, although information about the owner of a transaction is not revealed in the public blockchain address, this does not guarantee complete user privacy.

In addition, the immutability of smart contracts makes any error in coding it very dangerous as was brought to light by the DAO hack [29] on the Ethereum blockchain. With smart contract promising enforcing contracts on the blockchain, there is the question of whether such contracts would also be legally binding by default. Other challenges discussed in [33] include vulnerability to selfish miners, weakness of current consensus mechanisms and miners' centralization.

3 Related Works

Mišura and Žagar [21] described a model for a centralized data market place for IoT data. They envisioned a cloud service that allows sensor owners register their devices with relevant information about the sensor and the data it collects, and data consumers to query the system based on their requirement. Their platform architecture addresses publishing and discovery using a centralized device registry. Data from the sensor owners are first stored in a central measurement database to provide efficient delivery to multiple consumers as well as caching. Both the sensor owners and consumers interact with the platform using HTTP requests. They also reported on a system for ensuring data providers remain honest by evaluating the number of completed measurements divided by the number of agreed measurements. This measure is visible to the data consumers. Details of how the monetization would be implemented was not provided. Also, since their implementation is centralized, it introduces security and privacy issues.

Robert et al. [26] proposed a generic framework for data monetization also focusing on IoT data. The authors discussed considerations necessary for the design of a framework based on a P2P architecture. They outlined a few key requirements that such a platform should have including enabling information publication and discovery, secure money transactions, encouraging competitive pricing, using open standard-based platform, open market, and incentive for data sharing. They also analyzed some existing state of the art platforms such as Thingful (thingful.net), and concluded that none of them meets all the requirements. Still, a concrete design and/or implementation was not provided.

In [23], Noyen et al. proposed Bitcoin as a protocol for Sensing as a Service (S2aaS). The authors identified three challenges in this space: sensor identification (uniquely identifying and authenticating sensor owners), sensor data provenance (tracing sensor data and securing from manipulations) and low-cost micropayment (incentives for sensor owners to share data). They also identified some characteristics of the Bitcoin blockchain protocol that made it suitable for S2aaS applications including decentralization, pseudonymity, and low fees. A prototype of the idea was implemented by Wörner and von Bomhard [31]. The prototype consisted of three components: a sensor client, sensor repository and a requester client. The sensor client was implemented as a web socket to know when a payment has been made to the sensor address and it responds by addressing and publishing a transaction to the blockchain for the data consumer containing the requested data. The sensor repository was a centralized database with RESTful HTTP API and web interface allowing the sensor requester to search for desired sensor datasets. The authors identified a few challenges with their implementation including that the sent data will be publicly readable by every participant on the blockchain and scaling issues as more data is exchanged and stored on the blockchain.

Similarly, Wörner [30] developed a prototype for a decentralized market place for the exchange of data based on the 21 Bitcoin computer. Their motivation was to "free" sensor's data which they argued is "trapped in application-specific environments" by providing financial incentives to share data. Their implementation also provided means of discovery using a centralized sensor registry based on a MongoDB database.

Xu et al. [32] discussed a prototype of a platform for data monetization using smart contracts. They considered two scenarios: one where the data owner publishes their data to the platform and the data consumer browses for and select desired data set and the other, where the data consumers first pushes their jobs to the platform and the data owner can select jobs to provide data for. In both scenarios, the data owner is compensated for their data. The platform addresses the requirement of publishing and discovery using smart contracts, for example a dataset is registered by calling a dataset registry contract which stores description of the dataset along with a hash of the data; micro-payment infrastructures provided by the underlying blockchain cryptocurrency and provenance data is written for every event to the blockchain. The actual data is stored in an off-chain storage platform due to the size. The authors also pointed out the need for a reputation and rating mechanism to ensure that the data owners remain honest especially in describing their dataset. The fact that the data is centrally stored introduces the possibility of surveillance and data breaches. In addition, a clear idea of how it would be implemented, including evaluation was not provided.

4 Platform Architecture

The goal of this work is to propose, design, implement and evaluate an architecture on top of blockchain technology for publishing, discovery and exchange of data and services for cryptocurrency. The proposed system is called Cowry; it is a decentralized data and services monetization platform built on top of a blockchain providing users ability to trade their IoV/IoT data and services.

4.1 Design Considerations

In the design of the Cowry platform, we made certain design choices for the architecture, including how cryptocurrencies, encryption and keys selection was implemented.

Keys: Every account holder requires three different sets of keys:

- **Account key,** A_k**:** This is a public/private key pair generated by the underlying blockchain and used to sign every transaction made by the account. The account holder blockchain address is derived from the public key.
- **Encryption key,** E_k**:** This is a 32-bit symmetric key that is provided by the buyer each time a resource is to be shared. It is used to encrypt the transaction details as well as the resource (or information about accessing it) before writing it to the blockchain. The hash of this key is also required to retrieve the resource from the blockchain after the transaction is complete.
- **Sharing key,** S_k**:** This is a public/private key pair used to secure the transaction before it is completed. The encryption key, E_k is encrypted with the S_k public key of the data owner so it can use the private key to decrypt it. The reasons we chose to use a different key-pair for sharing data and for authenticating transactions are: (1) The account keys are built from Elliptic Curve Digital Signature Algorithm (ECDSA) [16] in MultiChain blockchain

platform and their primary purpose is for digital signatures. (2) Since the Account key is tied to the user account, it is more secure to support new encryption keys for each data exchange instead of reusing the account key for all transactions.

Hashing and Encryption: Hashing and encryption are very important elements in our architecture. Because of the open nature of the blockchain, everything published on it is visible to all network participants. To avoid this, we encrypted sensitive information such as the resource shared before writing to the blockchain ledger. We used AES (Advanced Encryption Standard) algorithm (symmetric encryption) with a key size of 256 bits and CBC (Cipher Block Chaining) encryption mode. For hashing, we used the SHA256 hashing algorithm.

Cryptocurrency: The platform uses it own cryptocurrency independent of that of the underlying blockchain native currency. The platform digital coin is called cowrie (plural: cowries). It is traded for exchange of digital resources. Using a currency different from the native currency of the blockchain platform helps to make the architecture independent of the underlying blockchain platform.

4.2 Architecture

The proposed solution serves as a middleware providing operations on top of the blockchain to utilize the infrastructure for exchanging data and services for cryptocurrency. Figure 1 shows a high-level overview of our proposed architecture.

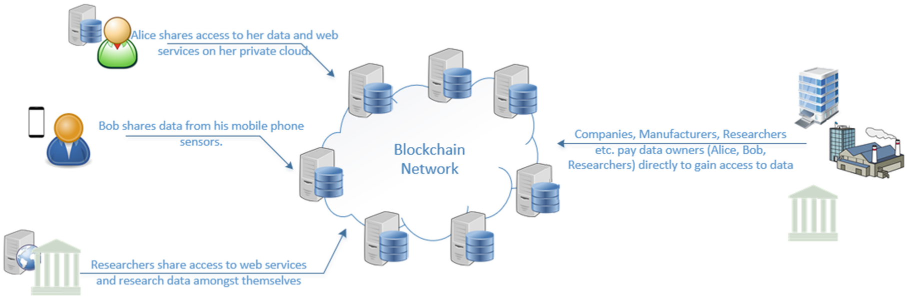

Alice shares access to her data and web services on her private cloud.
Bob shares data from his mobile phone sensors.
Researchers share access to web services and research data amongst themselves
Blockchain Network
Companies, Manufacturers, Researchers etc. pay data owners (Alice, Bob, Researchers) directly to gain access to data

Fig. 1. High-level overview of Cowry architecture

The blockchain network (shown in Fig. 1) consists of a number of Cowry nodes connected to each other to form a decentralized P2P network. Internally, each Cowry node has three layers (shown in Fig. 2): application layer, middleware layer and blockchain layer.

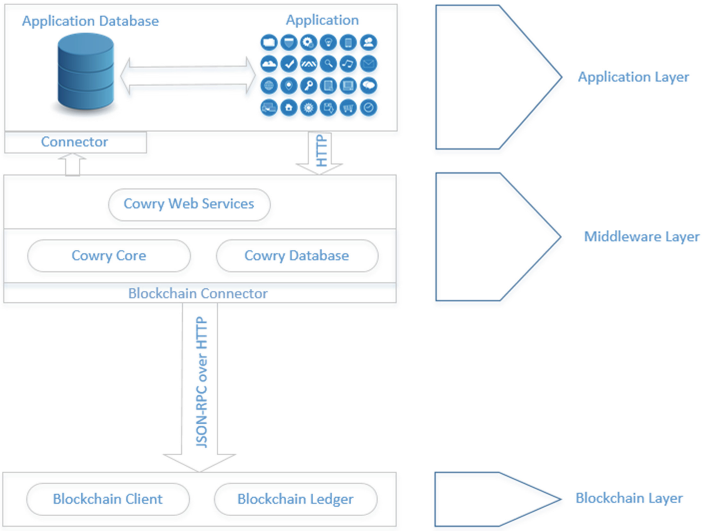


Fig. 2. Architecture of Cowry nodes

Application Layer: The application layer consists of application specific components and database. For example, this layer could contain an application running on a smart car along with its internal database which holds sensor data collected from the car. The application can connect to the Cowry core using RESTful web services to publish the data for trading.

Middleware Layer: The middleware layer consists of four components: Cowry Core, Blockchain Connector, Cowry Web service and Cowry Database.

- **Cowry Core:** The Cowry core defines a set of operations that enables the underlying blockchain to be used as a decentralized data and service marketplace. There are nine core operations provided by the Cowry middleware:
  - **Buy:** This allows the data consumer to request for a resource (data or service) published on the platform from the data owner. It requires the buyer's address as well as the resource unique identifier.
  - **Sell:** This allows the data owner to automatically respond to a buy request if it meets the price advertised. It is called automatically with the transaction data of the buy request.
  - **Rate:** This allows the data consumer to publish a rating between 0 and 1 for a transaction. It is called with details about the rating including the rated transaction id, the purchased resource id and the data consumer's comment.
  - **Search:** This allows a user to query the available data or jobs on the platform. The query is run on the node's local database and not on the blockchain.

  - **Sync:** This is used to synchronize the blockchain with the Cowry database. This is triggered automatically every time a new resource or job is published.
  - **Retrieve:** This is used to retrieved purchased resource from the blockchain. It is called with the symmetric key used for the transaction.
  - **View:** This allows a user to view the users, data, or jobs on the platform.
  - **Purge:** This purges the local database (Cowry Database) of expired entries. These entries are still available on the blockchain because of the immutability of the blockchain.
  - **Register:** This allows a user to register an account, a dataset or job on the platform. It is called with a JSON object containing different fields of information that can be published without encryption (thus visible to all) on the blockchain.
- **Blockchain Connector:** The blockchain connector helps to achieve modularity and a bit of independence from the underlying blockchain infrastructure ensuring the possibility of implementing the same architecture for different blockchains simply by using different blockchain specific connector. Also, this decoupling ensures that majors changes to the blockchain implementation does not require changing Cowry core.
- **Cowry Web Service:** The Cowry web service projects the operations of the Cowry core as services to be consumed via HTTP requests.
- **Cowry Database:** This component represents the unique part of this work. The Cowry database provides additional features for the platform that is not currently efficient using the underlying blockchain for example indexing for quick search and retrieval. The database caches some core data on the blockchain and index it for quick search. This is a trade-off between additional space (for the database) and performance improvement. A document style database was used because it natively supports the JSON format used by Cowry for storing the data on the blockchain.

Blockchain Layer: This layer contains the blockchain core client and local copy of the blockchain ledger. The blockchain ledger records all the transaction on the blockchain network. It was used to record all the interactions between participants, store the metadata of the participants and the metadata of the resource exchanged. As already mentioned, all data on the ledger is replicated on every participating node in the network. The blockchain client used for our prototype is MultiChain.

4.3 Sequence Diagrams

Figure 3 shows the process of buying and selling on the Cowry platform. The buy transaction is initiated by a user interested in a resource published on the Cowry platform. The sell transaction is initiated automatically by the seller's node if the buy requests meets the predetermined specification (e.g. price).

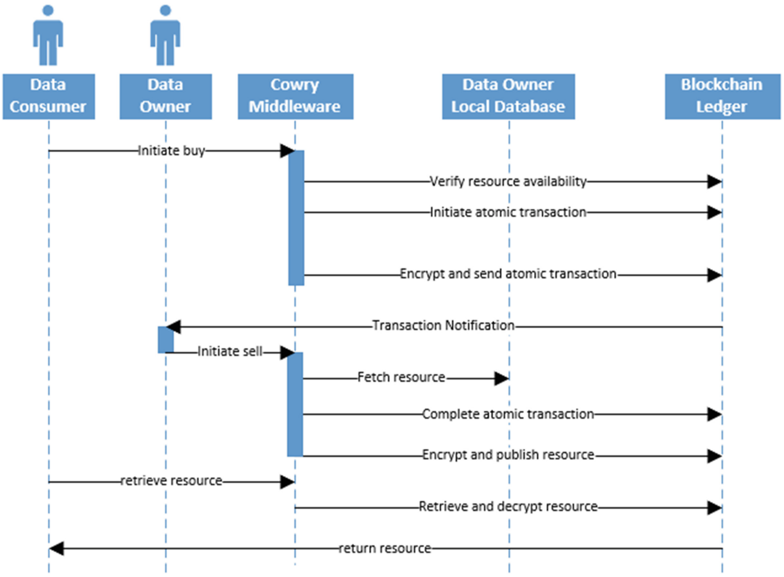

Data Consumer
Data Owner
Cowry Middleware
Data Owner Local Database
Blockchain Ledger
Initiate buy
Verify resource availability
Initiate atomic transaction
Encrypt and send atomic transaction
Transaction Notification
Initiate sell
Fetch resource
Complete atomic transaction
Encrypt and publish resource
retrieve resource
Retrieve and decrypt resource
return resource

Fig. 3. The process of buying and selling on Cowry platform

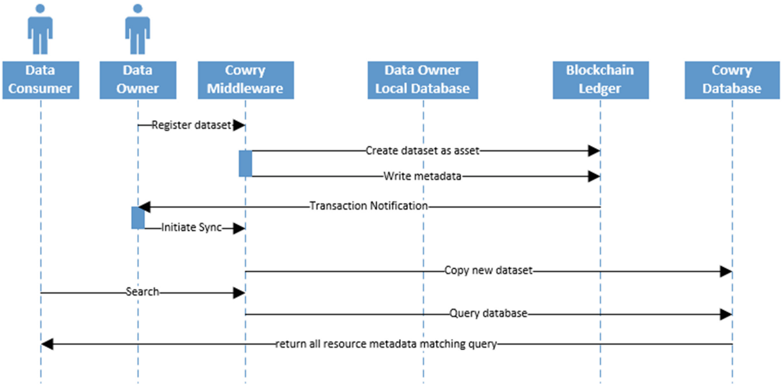


Fig. 4. The process of syncing and searching on Cowry platform

Figure 4 shows the Sync and Search process. Once the resource is uploaded and the metadata saved on the blockchain, it triggers an oracle on all the participating nodes which saves it to the Cowry database of that node. The metadata can then be queried with values for its different fields. For example, a user can query for dataset for a *location*, or *type*. The query goes directly to the Cowry database instead of the blockchain.

5 Evaluation

A prototype of the Cowry platform was built on top of a MultiChain private blockchain with 3 nodes on Amazon Web Services (AWS), each having the following specification:

- Amazon Instance type t2.xLarge
- Ubuntu 16.04 LTS (64-bit) OS
- Intel Broadwell E5-2686v4 @ 2.3 GHz
- 16 GB RAM

Two metrics were used for determining the system performance: Throughput and Response Time.

- **Response Time:** This is the time between when the client sends the request and when it receives the response. It is measured in milliseconds (ms).
- **Throughput:** This is the maximum rate at which requests is handled by the platform. It is calculated as:

$$number\ of\ requests/unit\ of\ time$$

where the time is measured from the start of the first request to the end of the last requests. It is measured in requests/seconds.

The response time acts as a proxy for measuring the user experience as a slow application lead to poor user experience. The throughput is a proxy for the scalability of the platform by capturing the performance of the system as the number of requests. Apache JMeter was used for the measurements. It is an open source testing tool used for testing the performance of a variety of services, including web services. The experiments would help determine the feasibility of building a fully decentralized data monetization platform based on the Cowry architecture.

The evaluation of the architecture focuses on the two primary operations introduced in this work:

- **Sync:** This operation ensures that the Cowry database and the blockchain are synchronized.
- **Search:** This operation allows flexible query of the resources metadata stored on the blockchain (and cached on the Cowry database).

Two experiments were performed:

1. Varying the number of users simultaneously accessing the platform.
2. Varying the delay between users' requests.

Experiment 1 - Varying the Number of Users Simultaneously Accessing the Platform: This experiment measures the average response time and throughput of the Sync and Search operation, while varying the number of users making the requests. The number of users is 1, 2, 5, 10, 20, 50 and 100. The average request is taken over 10 iterations.

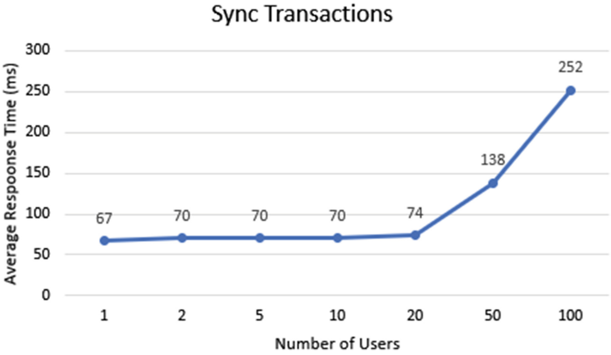


Fig. 5. Average response time for Sync transactions

Figure 5 shows the average response time for the synchronization operation on a single user node at 67 ms. This value rises to over 250 ms for Cowry nodes supporting up to 100 users. The value indicates the time it takes the local Cowry database to synchronize with the blockchain as it is updated with data by up to 100 users. The implication is any search operation conducted from that node within that period before the Cowry database synchronizes would not give the current state of the blockchain.

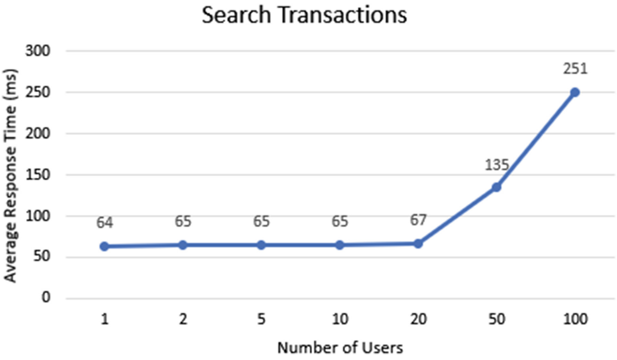


Fig. 6. Average response time for search transactions

Figure 6 shows the search operation on a single user node has an average response time of 64 ms. The search was conducted for dataset from a particular location and of a particular type. The first observation is that the average response time for the sync and search transactions follows the same pattern. It is fairly constant at about 67 ms for the up to 20 users before rising sharply to 251 ms for 100 users. Nevertheless, the speed of the search also depends on the amount of data to be retrieved. A more detailed experiment using standard benchmarks would be required to effectively test the performance of the search operation.

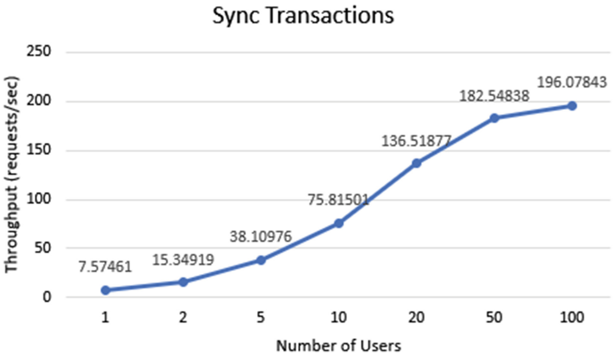


Fig. 7. Throughput for Sync transactions

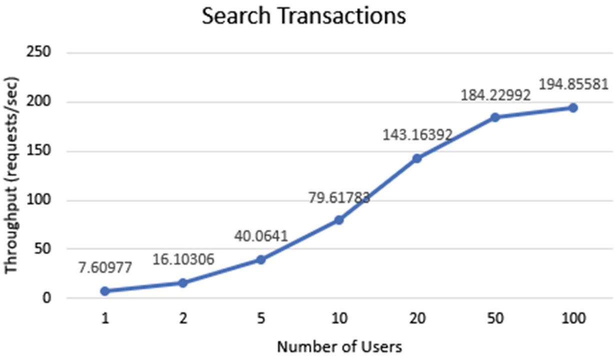


Fig. 8. Throughput for search transactions

The throughput for the sync transaction (Fig. 7) of the node supporting only a single user rises from about 8 requests per seconds to almost 200 requests per seconds for nodes supporting up to 100 users making simultaneous requests. This trend is very similar to that of the search transaction (Fig. 8) which also rises from about 8 requests per seconds to about 190 requests per seconds. This result suggests that the platform scales well.

Experiment 2 - Varying the Delay Between Users' Requests for 100 Users: In this experiment, the delay between the users' requests was varied from 250 ms to 1000 ms (with steps of 250 ms) for 100 users. Again, the average request is taken over 10 iterations (Figs. 9 and 10).

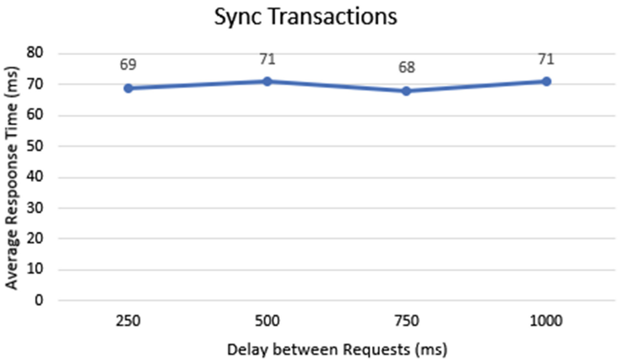


Fig. 9. Average response time for Sync transactions - 100 users

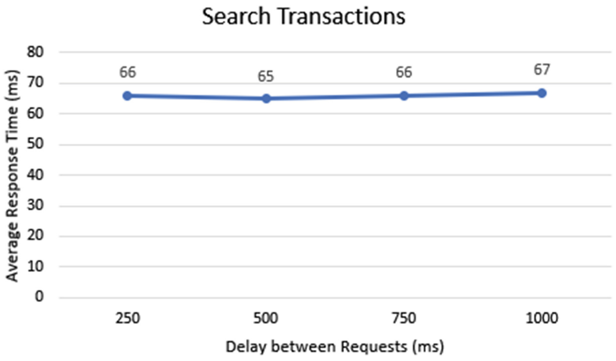


Fig. 10. Average response time for search transactions - 100 users

The average response time for the Sync and Search transaction on a node supporting 100 users making requests with delays from 250 ms to 1000 ms is fairly constant. This result suggests that the delays does not impact on the performance. However, the throughput measurement shows a fairly regular fall in the number of requests per seconds for both the sync and search transactions as the delay between requests is increase from 250 ms to 1000 ms. This is reasonable as the delays impacts on the total time used for computing the throughput (Figs. 11 and 12).

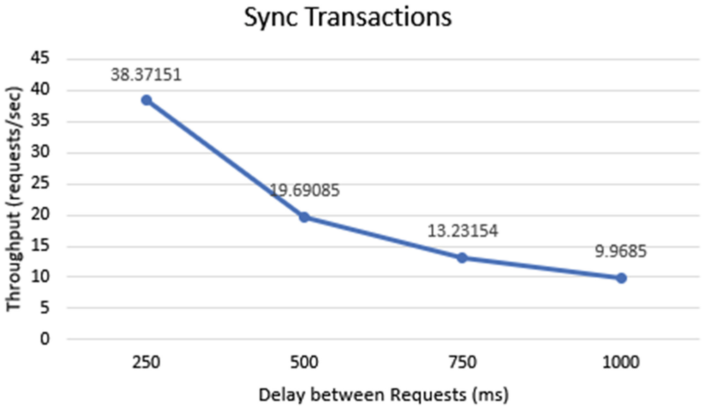


Fig. 11. Throughput for Sync transactions - 100 users

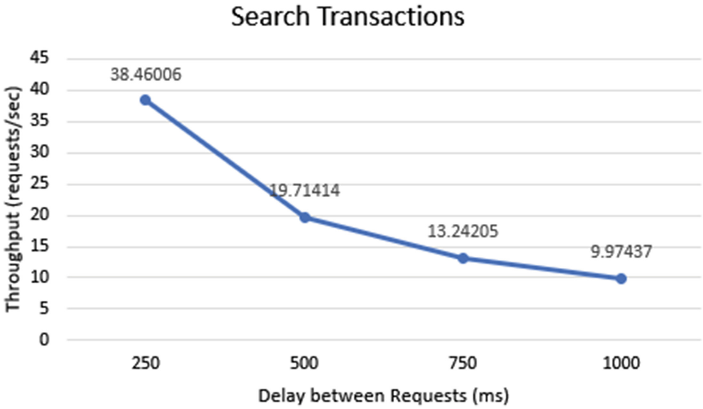


Fig. 12. Throughput for search transactions - 100 users

6 Conclusion

Two required features of a data monetization platform are query and retrieval of the metadata of the resources to be monetized. Centralized platforms rely on the maturity of traditional NoSQL database systems to support these features. These databases for example MongoDB allows for very efficient query and retrieval of data it stores. However, centralized platforms come with a bag of security and privacy concerns, making them not the ideal approach for a data monetization platform. On the other hand, most existing decentralized platforms are only partially decentralized. They, for example, leverage blockchain technology for its support of cryptocurrency and micropayments and some of its other features but still default to storing the traded resources metadata on a centralized database.

From our experiments, it can be concluded that the blockchain synchronizes with the Cowry database, and the metadata can be searched at very low latency, indicating the scalability and efficiency of the architecture. By adopting a modular implementation, any more efficient NoSQL document-style database can be used instead of MongoDB and would still accomplish the same or better results. The experiments also indicate that database features of robust and flexible query and retrieval can be leveraged alongside the decentralization benefits of the blockchain. The framework implemented in this work handles this by fast local synchronization between the blockchain ledger and a traditional NoSQL database system. The downside to the architecture is the additional space required to store the synchronized data.

A future work would be to explore how other blockchain platforms can be combined with traditional database systems with the aim of leveraging the database matured features for building a data monetization platform. Some of the blockchain platforms to investigate include Ethereum and Hyperledger fabric, both of which supports smart contracts.

References

1. Alam, K.M., Saini, M., El Saddik, A.: Toward social internet of vehicles: concept, architecture, and applications. IEEE Access **3**, 343–357 (2015)
2. Bahga, A., Madisetti, V.K.: Blockchain platform for industrial Internet of Things. J. Softw. Eng. Appl. **9**, 533–546 (2016). http://www.scirp.org/journal/jsea
3. Bitcoin Wiki: Double spending. https://en.bitcoin.it/wiki/Double-spending. Accessed 28 Mar 2017
4. Carboni, D.: Feedback based reputation on top of the Bitcoin blockchain. arXiv preprint arXiv:1502.01504 (2015)
5. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the Internet of Things. IEEE Access **4**, 2292–2303 (2016)
6. Coin Sciences: MultiChain - open platform for blockchain applications. https://www.multichain.com/. Accessed 28 Mar 2017
7. Coin Sciences: MultiChain permissions management. https://www.multichain.com/developers/permissions-management/. Accessed 10 Aug 2017
8. Coin Sciences: MultiChain runtime parameters. https://www.multichain.com/developers/runtime-parameters/. Accessed 10 Aug 2017
9. Finley, K.: Tim Berners-Lee, inventor of the web, plots a radical overhaul of his creation. https://www.wired.com/2017/04/tim-berners-lee-inventor-web-plots-radical-overhaul-creation/. Accessed 4 Aug 2017
10. Fischer, M.J.: The consensus problem in unreliable distributed systems (a brief survey). In: International Conference on Fundamentals of Computation Theory, pp. 127–140. Springer, Heidelberg (1983)
11. Fox-Brewster, T.: Ashley Madison breach could expose privates of 37 million cheaters. https://www.forbes.com/sites/thomasbrewster/2015/07/20/ashley-madison-attack/#85148765f48c. Accessed 30 June 2017
12. Goel, V.: Facebook tinkers with users emotions in news feed experiment, stirring outcry. https://goo.gl/HYDba4. Accessed 26 May 2017
13. Greenspan, G.: Introducing multichain streams. http://www.multichain.com/blog/2016/09/introducing-multichain-streams/. Accessed 28 Mar 2017
14. Greenspan, G.: MultiChain data streams. http://www.multichain.com/developers/data-streams/. Accessed 28 Mar 2017

15. Greenwald, G., MacAskill, E.: NSA prism program taps in to user data of Apple, Google and others. https://www.theguardian.com/world/2013/jun/06/us-tech-giants-nsa-data. Accessed 5 June 2017
16. Johnson, D., Menezes, A., Vanstone, S.: The elliptic curve digital signature algorithm (ECDSA). Int. J. Inf. Secur. **1**(1), 36–63 (2001)
17. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of Bitcoins: characterizing payments among men with no names. In: Proceedings of the 2013 Conference on Internet Measurement Conference, pp. 127–140. ACM (2013)
18. Merkle, R.: A digital signature based on a conventional encryption function. In: Advances in Cryptology CRYPTO 1987, pp. 369–378. Springer, London (2006)
19. Microsoft: Networking basics: peer-to-peer vs. server-based networks. https://technet.microsoft.com/en-us/library/cc527483(v=ws.10).aspx. Accessed 27 July 2017
20. Microsoft: Understanding public key cryptography. https://technet.microsoft.com/en-us/library/aa998077(v=exchg.65).aspx. Accessed 28 Mar 2017
21. Mišura, K., Žagar, M.: Data marketplace for Internet of Things. In: International Conference on Smart Systems and Technologies (SST), pp. 255–260. IEEE (2016)
22. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008)
23. Noyen, K., Volland, D., Wörner, D., Fleisch, E.: When money learns to fly: towards sensing as a service applications using Bitcoin. arXiv preprint arXiv:1409.5841 (2014)
24. Ouaddah, A., Elkalam, A.A., Ouahman, A.A.: Fairaccess: a new blockchain-based access control framework for the Internet of Things. Secur. Commun. Netw. **9**(18), 5943–5964 (2016)
25. Perera, C.: Sensing as a service (S2aaS): Buying and selling IoT data. arXiv preprint arXiv:1702.02380 (2017)
26. Robert, J., Kubler, S., Le Traon, Y.: Micro-billing framework for IoT: research & technological foundations. In: IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), pp. 301–308. IEEE (2016)
27. Robinson, T.: Bitcoin is not anonymous. http://www.respublica.org.uk/disraeli-room-post/2015/03/24/bitcoin-is-not-anonymous/. Accessed 1 Aug 2017
28. Sharples, M., Domingue, J.: The Blockchain and Kudos: a distributed system for educational record, reputation and reward. In: European Conference on Technology Enhanced Learning, pp. 490–496. Springer, Cham (2016)
29. Siegel, D.: Understanding the DAO attack. https://www.coindesk.com/understanding-dao-hack-journalists/. Accessed 11 Aug 2017
30. Wörner, D.: Design of a real-time data market based on the 21 Bitcoin computer. In: Proceedings of the 11th International Conference on DESRIST 2016, St. Johns, 23–25 May 2016, pp. 228–232. Springer, Cham (2016)
31. Wörner, D., von Bomhard, T.: When your sensor earns money: exchanging data for cash with Bitcoin. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 295–298. ACM (2014)
32. Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A.B., Chen, S.: The blockchain as a software connector. In: 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), pp. 182–191. IEEE (2016)
33. Zheng, Z., Xie, S., Dai, H.N., Wang, H.: Blockchain challenges and opportunities: a survey (2016)

A Blockchain-Based Traceable Certification System

Po-Yeuan Chang[1], Min-Shiang Hwang[2,3](✉), and Chao-Chen Yang[1]

[1] Department of Management Information Systems, National Chung Hsing University,
Taichung, Taiwan, R.O.C.

[2] Department of Computer Science and Information Engineering, Asia University,
Taichung, Taiwan, R.O.C.
mshwang@asia.edu.tw

[3] Department of Medical Research, China Medical University Hospital,
China Medical University, Taichung, Taiwan, R.O.C.

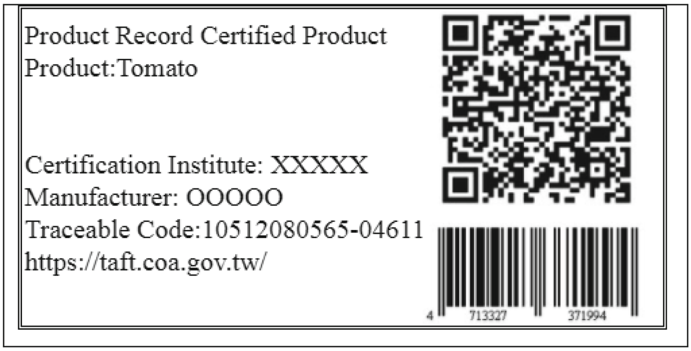
Abstract. In recent years, product records become more common to merchandize sold in retail stores, but the current product record system used today can't assure product's quality after products were transported through the whole supply chain. During transportation, merchandise may be damaged accidentally or condition changed. Those events do not get recorded because records are predominantly focused by manufacturers. Also in second hand market, product record may be tampered or verification is weak. Nonexperiences buyers can't distinguish counterfeit because records are not trustworthy and outdated. By using the concept borrowed from Bitcon, the advantage of the blockchain can be applied to the product record system. Because of the characteristics of the blockchain such as: decentralization, openness, and immutability, which can improve the system. To achieve the goal, ownership of products is introduced and smart contract is also embedded to further enhance the product record system.

Keywords: Blockchain · Product record · Ethereum · Security

1 Introduction

Counterfeits have caused both financial lost and may even damage company's brand reputation in the current fast paced global market. To prevent counterfeit, manufacturers currently use a "Product Record" which is a kind of physical sticker containing digital information such as: Where and when a product was made, address of the manufacturer, verification from a fair third party institution, etc. QR code (Quick Response Code) [1] is widely used on brand product as a quick and convenient way to access product record by consumer. When consumers scan a QR code picture using a mobile device, product record stored in database will pop up instantly [2, 3]. Product record is extensively used in the agricultural products and high value brand products. The former is to prevent pesticide residue due to food safety; and the latter is to prevent counterfeit. Figure 1 shows an example of "agriculture production & selling verified record" which published by Council of Agriculture and this picture contains product name, verified institution, traceable serial number, web page link and QR code. By scanning QR code, product information such as famer name, package date, nutrition table, and even the whole planting schedule can be

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 363–369, 2018.
https://doi.org/10.1007/978-3-319-76451-1_34



Product Record Certified Product
Product:Tomato

Certification Institute: XXXXX
Manufacturer: OOOOO
Traceable Code:10512080565-04611
https://taft.coa.gov.tw/

4 713327 371994

Fig. 1. Example of agricultural product record

seen. In Fig. 1, QR code on the left will direct consumer to a webpage site (https://taft.coa.gov.tw/) where details of this box of tomatoes can be seen.

Information of product stored in an authorized institution's database is difficult to update simultaneously during the process of shipping to distributer and retailer. Currently, the system used is predominantly focused only on the manufacturer side, but what happened during a supply chain will be ignored easily. Also a second-hand market may have to rely on experience to distinguish whether the product is counterfeit or not. The chance for those without experience or technique, to buy accidentally counterfeits will increase. Thus the method using a blockchain based technique is introduced to prevent situations above.

2 Preliminaries

Before implementing the method, some preliminaries should be established first. Blockchain is a continuously growling list of record served as an open, distributed ledger. Due to its distinct characteristics, digital currency like Bitcon [4, 5] can be developed. Take Bitcoin as an example: each Bitcoin block contains timestamp, previous block hash, transaction data, and nonce value. It can be considered as a block of message holding information that describes the current status. Figure 2 is a simple organization of Bitcoin.

Blockchain has following distinct characteristics to make it unique and interesting [6]:

A. Decentralization

The advantage of decentralization is that third party authorization is not needed. Data can be stored across blockchain network, so the risk of holding information centrally can be minimized [7]. If one of nodes in blockchain loses its data, the rest of nodes always have a copy to fix this problem.

B. Openness

All nodes of the blockchain network have to be synchronized before joining a network. Every user transaction can be traced back, so users can see what transactions have done before. Many people and companies dedicated to the improvement and progression of

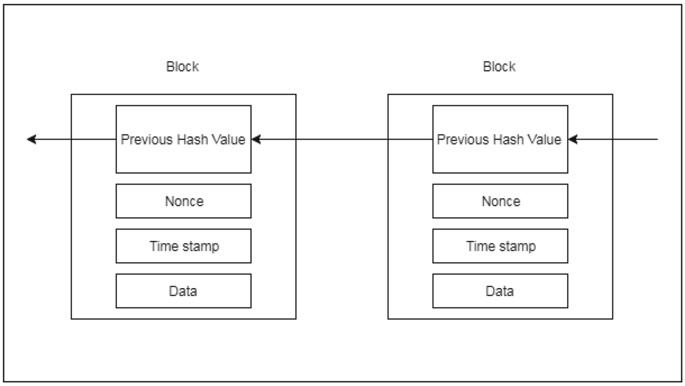


Fig. 2. Organization structure of blockchain

blockchain. Software engineers can use source code opened online to develop any application they need.

C. Immutability

Before blockchain is formed, all nodes who participate in the blockchain system have to guess the nonce value of message data. This action is called "Mining" and considered as a proof of work. Just like lottery, these activities come with reward. If one miner is lucky enough to guess the right nonce value, the miner can get rewarded with benefit. After it is mined, a new block is created and added to blockchain. Therefore, the messages packed into blockchain are nearly impossible to alter since malicious attackers have to mine the whole blockchain from the starting block to the latest one.

3 Implementing System

In this section, process of implementing a blockchain based product record system will be explained by using an example.

The system has to fit following requirements to obtain a safe and efficient usable system.

1. Open Data

Users can search information of product they own or are going to buy easily without any restrictions. For both merchant and costumer, openness means the fair trading condition which is beneficial to an open market. The more information a user can reach the more trust manufacturer can obtain from user.

2. Tamper Resistant

Data can't be tampered to ensure correctness of information. If data can be changed by an attacker, the foundation of product record will collapse. Therefore, it is crucial that information is in a safe keeping condition. This requirement can be achieved by implementing blockchain, because each block has a hash number which points to previous block and every node have a copy of blockchain to ensure it is decentralized. These characteristics make hostile tampering impossible [8, 9].

3. Record Traceability

All transaction should be traceable to ensure the product supply source is legit. Counterfeits may be blended into a real product in the supply chain if a retailer can't tell where product was made and when it was shipped. Timestamps in blockchain can be used to keep supply chain intact, thus leaving no chance to counterfeits.

4. Decentralized

The system can run without central institution authorization but it still can be trustworthy. This is especially important when most second hand product don't have solid verification to prove the product condition.

After knowing these essential terms above, the system can be built sequentially [10–17] and system is shown in Fig. 3 below.

(1) Products are made

When a new product P is manufactured in factory F. Information such as company prefix, product series number, manufacture time & address and ownership will be broadcasted to every node in the Ethereum network. Then nodes start to mine a nonce value. After being mined and approved by other nodes, message M1 containing information above will be packed into a block. Note that only brand licensed factories can claim the initial product ownership.

(2) Products are distributed

When product is shipped to distributor D, transaction between manufacturer and distributor will occur. First a distributor checks product's ownership via the Ethereum platform to ensure what he received is not counterfeited. Then the message "Product checked & received" will be sent to Ethereum and the ownership of P is transferred from F to D. The transaction information M2 will also be broadcasted to every node in Ethereum for further blockchain mining. In block M2 there will be a pointer that points to previous block M1. After the shipment is done, block M2 will be connected to previous block M1, which form a simple blockchain. Both M1 and M2 can represent P's condition, so "Product Record of P" is formed. In each stage of the product supply chain distributors and retailers can add new information to transaction blocks.

(3) Products are bought by customers

Customers in store can check P's ownership and product record on the Ethereum network before buying. Furthermore, when costumers decide to sell P to a second hand shop or pawnshop, product record can prove that it is from legit manufacturer through the qualified supply chain.

Product manufacturers and retailers are blockchain miners in product record system. As nodes in Ethereum, factories and retailer create a private chain which is formed by blocks containing product record. By implementing private chain, the cost of each block can be reduced and also provides a costumer friendly environment.

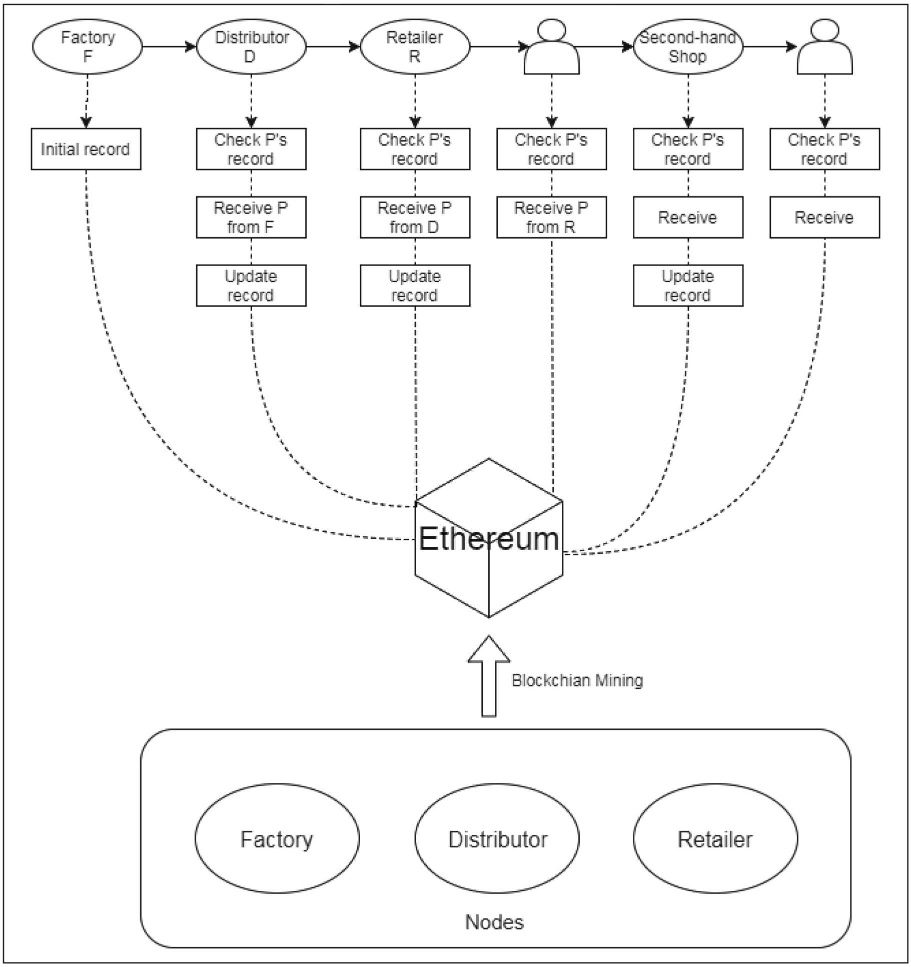


Fig. 3. Example of a blockchain based product record system

For instance, a France company which is famous for its fashion design purse decides to launch its new product of classic series. This high-end product is anticipated by consumers around the world. Counterfeits are foreseen to appear in the first week after the launch day. A licensed factory in China will first claim the initial ownership of product made. Then, the factory transfers its ownership to a cooperated transport company before the cargo is shipped to America. The purse will be delivered to retailers in the big city like New York and so does its ownership. While ownership of the purse is transferred to next stage of the supply chain, more information about this product can be added to the block. Consumer in store can view ownership record by using the Ethereum network to make sure the purse they buy is not counterfeited. When the purse is outdated, its owner can take it to a second hand shop hopping there still has a good price. The shop owner can check the product record of the purse to know when and where it was made to determine the price offered.

The major difference between traditional product record and blockchain embedded system is that events occurred during transportation or after product was sold can be recorded and traced back. Without third party institution verification the product is still worthy of trust because of block-chain's tamper resistant. To those active second hand market: automobiles, electronic products and high value goods, the blockchain based system can ensure both quality and safety.

The blockchain based system also provides companies a better way to trace and protect their products easily. Product managers can know where their products were shipped correctly since blockchain is hard to tamper. By tracing product record inside blockchain, maintenance engineers can find out information needed. A new product management model will be implemented due to the blockchain embedded product record system.

4 Conclusion

Thanks to the popularity of Bitcoin, Blockchain technology is widely used in e-commerce. It explores software engineering to a brand new area. Product record system is one of many subjects that can be improved by using blockchain. The main purpose of inventing product record is to protect consumers from being deceived. In modern society, security and efficiency are two major concerns of any new invented technology. Especially security of financial data includes not only personal information but also transaction data. Blockchain uses its distributed system to form a barrier that is hard to break. This concept can be applied to many different research fields.

Still there are some open problems in the blockchain based system. How to determine rewards to those who participate in mining? [18] If rewards are few, the motivation of mining may be lost. How to balance system cost and profit using this system? How to defense attack on Ethereum using DDoS [19–22] is another interesting issue. This system may provide a structure for the improved product record system, and yet more detail and practical operation can be further discussed.

References

1. QRcode.com: DENSO wave, the Inventor of QR code. https://www.qrcode.com/en
2. Qin, J., Sun, R., Xiang, X., Li, H., Huang, H.: Anti-fake digital watermarking algorithm based on QR codes and DWT. Int. J. Netw. Secur. **18**(6), 1102–1108 (2016)
3. Narasimhan, H., Padmanabhan, T.R.: 3CAuth - a novel multi-factor authentication scheme using QR-code. Int. J. Netw. Secur. **18**(1), 143–150 (2016)
4. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008). https://bitcoin.org/bitcoin.pdf
5. Ibrahim, M.H.: SecureCoin: a robust secure and efficient protocol for anonymous bitcoin ecosystem. Int. J. Netw. Secur. **19**(2), 295–312 (2017)
6. Lin, I.C., Liao, T.C.: A survey of blockchain security issues and challenges. Int. J. Netw. Secur. **19**(5), 653–659 (2017)

7. Zyskind, G., Nathan, O., Pentland, A.: Decentralizing privacy: using blockchain to protect personal data. In: Security and Privacy Workshops, pp. 180–184. IEEE (2015)
8. Liu, Y., Chang, C.C., Chang, S.C.: An efficient and secure smart card based password authentication scheme. Int. J. Netw. Secur. **19**(1), 1–10 (2017)
9. Moon, J., Lee, D., Jung, J., Won, D.: Improvement of efficient and secure smart card based password authentication scheme. Int. J. Netw. Secur. **19**(6), 1053–1061 (2017)
10. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
11. Charlon, F.: Openassets/open-assets-protocol: Technical specification for the open assets protocol, a bitcoin based colored coins' implementation (2013). https://github.com/OpenAssets/open-assets-protocol
12. Christidis, K., Devetsikiotis, M.: Blockchains and smartcontracts for the Internet of Things. IEEE Access **4**, 2292–2303 (2016)
13. Wood, G.: Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper (2014)
14. Loibl, A., Naab, J.: Namecoin. Netw. Archit. Serv. 107–113 (2014)
15. Delmolino, K., Arnett, M., Kosba, A.E., Miller, A., Shi, E.: Step by step towards creating a safe smart contract: lessons and insights from a cryptocurrency lab. IACR Cryptol. ePrint Arch. **2015**(460), 1–15 (2015)
16. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: anonymous distributed E-Cash from Bitcoin. In: IEEE Symposium on Security and Privacy (2013)
17. Sasson, E.B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: decentralized anonymous payments from Bitcoin. In: IEEE Symposium on Security and Privacy (SP), pp. 459–474 (2014)
18. Gas fees for ethereum operations. http://ether.fund/tool/gas-fees
19. Mirkovic, J., Reiher, P.: A taxonomy of DDoS attack and DDoS defense mechanisms. SIGCOMM Comput. Commun. Rev. **34**(2), 39–53 (2004)
20. Baishya, R.C., Hoque, N., Bhattacharyya, D.K.: DDoS attack detection using unique source IP deviation. Int. J. Netw. Secur. **19**(6), 29–939 (2017)
21. Behal, S., Kumar, K.: Characterization and comparison of DDoS attack tools and traffic generators: a review. Int. J. Netw. Secur. **19**(3), 383–393 (2017)
22. Behal, S., Kumar, K., Sachdeva, M.: Discriminating flash events from DDoS attacks: a comprehensive review. Int. J. Netw. Secur. **19**(5), 734–741 (2017)

Author Index

S.-L. Peng et al. (Eds.): SICBS 2017, AISC 733, pp. 371–372, 2018.
https://doi.org/10.1007/978-3-319-76451-1