Cloud-based Personal Data Protection System and Its Performance Evaluation

Jung-Chun Liu, Chu-Hsing Lin, Ken-Yu Lee

Department of Computer Science, Tunghai University, Taiwan
{jcliu, chlin}@thu.edu.tw, s2013453@gmail.com

Abstract

This article uses Information technology (IT) to assist in the fulfillment of personal information protection and reduce IT risks within an organization. Advanced IT approaches are adopted to locate and verify personal information. Whenever the Personal Information Protection Act (PIPA) is violated unnoticeably by users, the proposed system will effectively detect files with personal information by means of cloud computing and alert those users. This study uses Hadoop distributed computing platform to support computation of huge amount of data. To avoid the risk of information leakage when duplicating personal information to worker nodes in Hadoop cloud platform, personal information is hashed before transmission. To detect personal information, documents are analyzed using automata-based programming to locate suspicious words. Every suspicious word is then verified with the help of a personal information database. Finally, this study analyzes the computing efficiency of Hadoop nodes and experimentally shows how to adjust the number of maps in each node of the Hadoop MapReduce structure to optimize system performance.

Keywords: Automata, Cloud computing, Digital signature, Information security, Personal Information Protection Act

1 Introduction

Data privacy governs how the data is collected, shared and used; however, if it is collected without proper consent that is a violation of data privacy [1-2]. The Personal Information Protection Act (PIPA) has already been enacted by the Taiwanese government since 2010. However, its result is worse than expected. People merely have limited awareness about usages of personal information such as personal identification number, bank account number, date of birth, and medical data [3]. When transmitting documents, they might not be aware of disclosing personal information via text words or numbers, thus unwittingly violating PIPA. According to definition of personal information in Article 2 of PIPA [4], we first develop an automata-based system to analyze personal information, including personal name, address, and identification number to effectively scan and detect personal information in documents.

The proposed system consists of the server and client sides. The client retrieves personal information from documents. The retrieved personal information will be verified via digital signatures, filed for future reference, and uploaded to the server to perform personal information analysis.

The proposed automata-based system can be used in organizations to improve protection of personal information. It supports documents in various file formats, such as e-mail, Office document, webpage, and text file. Also, to accelerate computation speed of handling data of massive volumes, a cloud-based personal data protection system implemented with a Hadoop framework is further developed.

2 Background

2.1 Hadoop Cloud Computing

Apache Hadoop, an open-source framework, supports distributed data processing and storage of huge amount of information using computing nodes in a cluster [5]. Hadoop mainly consists of a processing component (i.e., MapReduce) and a storage component (i.e., Hadoop Distributed File System, HDFS). It implements the MapReduce programming model to split files into blocks that are distributed over computing nodes in the cluster [6-7]. Besides, it uses HDFS, the distributed file system, to store data processed in computing nodes, thus enabling fast and efficient data processing [8-9].

2.2 Automata-based Programming

The automata-based programming model is a programming paradigm in which the program or part of it is a model of a finite state machine (FSM). It has the following two features. First, the execution time of the program can be clearly divided into separate automata steps, each step consisting of a code section

*Corresponding Author: Chu-Hsing Lin; E-mail: Chlin@thu.edu.tw
DOI: 10.3966/160792642019102006004

with a single entry-point. Second, communication between steps is only possible through the explicitly noted variables set, i.e., the state. Automata-based programming is often used in lexical analysis and syntax analyzers. It is very useful for analyzing personal information. Based on words retrieved from sentences, the state is changed according to state conditions. When the final state is reached, the words are words that match set conditions [10].

2.3 Document Retrieval

To facilitate general applications, the proposed system contained retrieval tools for documents in various formats as listed in Table 1. For processing traditional documents in txt format, BufferedReader of JAVA was called. For processing Office documents, Apache POI [11] was used since it could provide pure Java libraries for reading and writing files in Office formats such as Word, PowerPoint, and Excel. We also included jsoup that could crawl and parse tags and texts in a webpage [12]. For analysis of users' e-mails, we used JavaMail API to provide a platform-independent and protocol-independent framework to build e-mail and messaging applications. After entering user passwords, JavaMail API was used to read the whole contents inside a mailbox, retrieve e-mails, and attach files for analysis.

Table 1. Types of documents and retrieval methods

Type of Document	Retrieval Method
Traditional documents	BufferedReader
Office documents	POI
Web page	jsoup
E-mail	JavaMail

3 Experimental Design

As shown in Figure 1, the proposed Hadoop system consists of client and server sides as described in detail below:

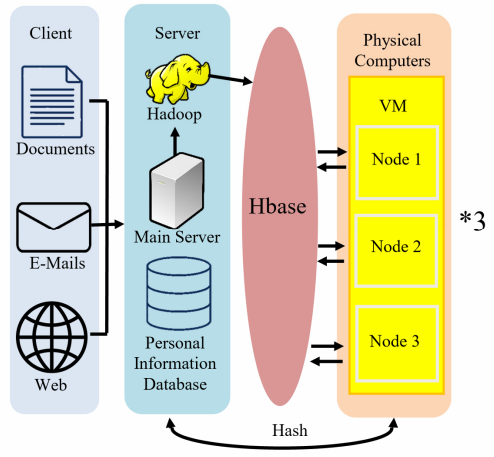

Figure 1. Architecture of the Cloud-based Personal Data Protection System using the Hadoop platform

3.1 Client

The client is in charge of retrieving documents classified into four types as listed in Table 1. The system automatically retrieves documents stored in the client. Besides, when users are transmitting data via webpages, the system will analyze contents of the transmitted data and warn users before transmitting sensitive personal information, thus greatly enhancing privacy of users. The system also offers protection for users' e-mail box. With permission from users, it can read and retrieve documents including e-mails and attached files in the mailbox. The retrieved document is certificated via digital signatures. After finding personal data on the server side to ascertain its source client, the retrieved personal data are filed for future reference. The document certificated by the digital signature will be uploaded to the main server on the server side to perform personal information analysis [13].

3.2 Server

The server side consists of the main server, database, and Hadoop. The main server is in charge of receiving documents and communicating with clients. The database comprises personal information database and backup database. The personal information database is used to verify if personal information filtered by the algorithms matches with any personal information inside the organization. Files containing personal information are stored in the backup database with records of their user names. Hadoop is in charge of executing main algorithms. Since the initial stage of operation of the proposed system involves massive amounts of documents, cloud computing technology is introduced into the system to speed up the overall system performance. However, Hadoop has a drawback in that it duplicates data to every node, i.e., matched personal information accessed from the personal information database is copied to nodes, thus increasing the risk of personal information leakage. To solve this problem, personal information is hashed before transmission as a means to enhance privacy preservation for cloud data storage [14-15].

3.3 Hadoop

Apache Hadoop, an open-source distributed cloud computing system, can quickly analyze huge amounts of data. It consists of a master node and multiple worker nodes [6, 16]. The number of nodes can be dynamically increased or decreased. The Hadoop system can process huge amounts of data in parallel by using large quantity of nodes. It offers very powerful operation performance. The Hadoop cloud architecture stores duplications of files in nodes. Hence, it occupies a lot of storage space and causes latency of network transmission. Different from GPU that can handle simple operations, Hadoop can execute complex codes

such as read and write access to database. However, since database is independent of Hadoop nodes, operations of database will be very time-consuming. It is critical to effectively reduce amounts of queries to database in Hadoop. The proposed system uses automata-based programming to analyze sentences to effectively reduce amounts of queries to database.

3.4 Algorithm

The proposed algorithm adopts automata-based programming to check and analyze words associated with personal names, addresses, and identification numbers that consist of data in non-fixed, part-fixed, and fixed formats, respectively.

3.4.1 Data in Non-fixed Format

For data in non-fixed formats, names in personal information database are arranged for analysis. The automata-based program used for name search compares words in sentences to find if a word matches with any stored Chinese last name in the personal information database. When a matched word is located, it sends the word next to the detected last name to the database. All similar names in personal information database are hashed and then transmitted to computing nodes which acquire one word at a time through the document, hash it, and then compare it with the received hashed word of similar names. The flow chart of the automata-based program used for name search is shown in Figure 2.

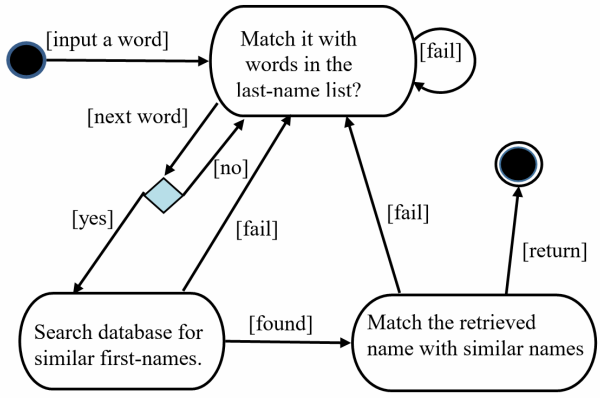

Figure 2. Flow chart of the automata-based program used for name search

3.4.2 Data in Part-fixed Format

For data in part-fixed formats, the fixed format part of them can be used for analysis. For example, an address typically consists of words “county”, “city”, “area”, “street”, and ends with the word “number” in a Chinese address. The flow chart of the automata-based program used for address search is shown in Figure 3.

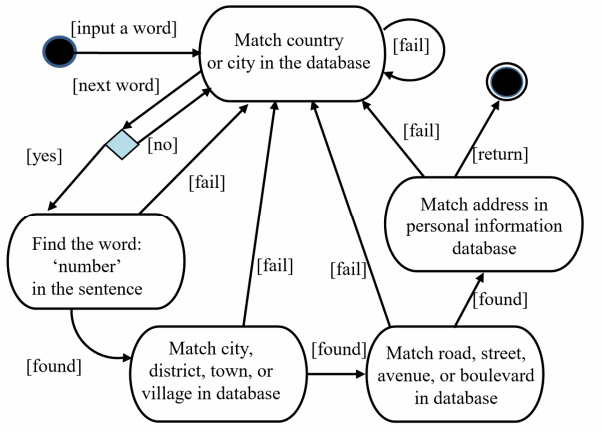

Figure 3. Flow chart of the automata-based program used for address search

3.4.3 Data in Fixed Format

Analysis of data in fixed format is relatively easy. It can be performed by analyzing words to see if contents of the expected fixed format are matched. For example, the personal identification number used in Taiwan consists of one English alphabet followed by 9 decimal numbers which always start with number 1 or 2. The flow chart of the automata-based program used for personal identification number search is depicted in Figure 4.

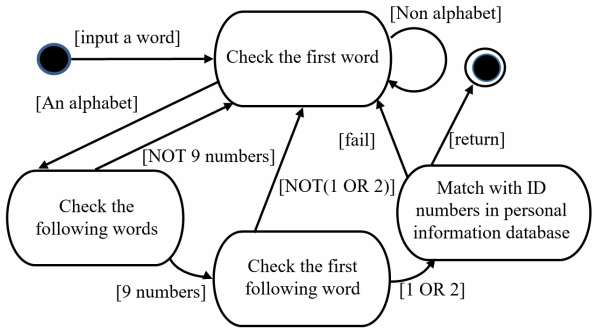

Figure 4. Flow chart of the automata-based program used for matching identification number

4 Experimental Design

In this study, first, we investigated the efficiency of the automata-based system implemented on a single PC, and second, based on the automata-based system framework, we introduced the Hadoop platform and performed performance comparison of the proposed Hadoop system by comparing its execution time with that of a standalone PC. Specifications of hardware and software of the experimental environment are listed below.

Hadoop system.

Master node: G860 CPU with 8G RAM.

Worker nodes: three physical computers, each with Core i7-4790 CPU and 8G RAM, in each one of them, three virtual machines (VMs) are created, each with one Core and 2G RAM to form 9 worker nodes.

Single PC.

A physical computer with Core i7-4790 CPU and 8G RAM is used, in which one VM with 1 Core and 2G RAM are created.

Personal information database.

The database contains 2,200 real personal information data such as name, personal identification number, address, telephone number, and student identification number.

4.1 Experiment 1

In this experiment, the execution time of the automata-based system implemented on a single PC to process various amounts of data is first investigated. The experimental data used consist of one-year national news in Taiwan. Each piece of news is duplicated 100 copies before it is stored in the database. Furthermore, the execution time of the proposed Hadoop system is compared to that of the single PC. Execution times for Hadoop systems (with 1 to 9 worker nodes) and single PC are listed in Table 2 and plotted in Figure 5. The data consisted of various pieces of news, ranging from 100,000 to 1,000,000 pieces.

Table 2. Execution times (in seconds) for processing various in pieces of news using Hadoop systems with various worker nodes or the single PC system

Data size (pieces)	1_node (sec)	2_node (sec)	3_node (sec)	4_node (sec)	5_node (sec)	6_node (sec)	7_node (sec)	8_node (sec)	9_node (sec)	Single PC (sec)
1 M	23946	16622	10032	9530	8119	8041	7428	6641	6477	13846
800K	18862	11222	8986	7460	6914	6627	5998	5173	5074	10774
600K	14102	9023	7305	5405	5128	4348	4078	3936	3759	8001
400K	9134	4903	4167	3938	3314	3075	2831	2721	2439	5053
200 K	4259	3119	1976	1800	1461	1326	1319	1264	1148	2212
100 K	2170	1390	1378	802	774	716	686	657	588	1118

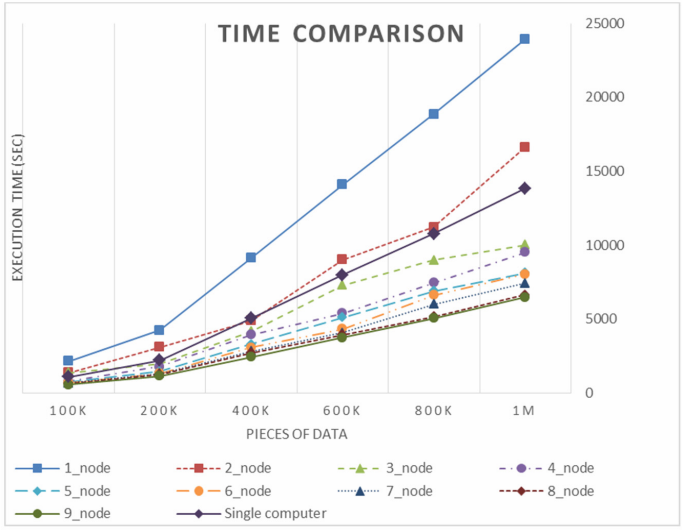

Figure 5. Execution times (in seconds) for processing various pieces of news using Hadoop systems (with various worker nodes) or the single PC system

As shown in Table 2 and Figure 5, the execution time of the automata-based system grows linearly as the data size increases. For the single PC system, it takes 18.6 minutes (1,118 seconds) to process data of 100K pieces and 230.8 minutes (13,846 seconds) to process data of 1M pieces. For the proposed Hadoop system, it takes 9.8 minutes (588 seconds) to process data of 100K pieces and 108.0 minutes (6,477 seconds) to process data of 1M pieces.

Speedup ratios of Hadoop systems (with 1 to 9 worker nodes) over the single PC system are listed in Table 3 and plotted in Figure 6. Experimental results revealed that the performance of the Hadoop system was worse than the single PC system when worker nodes used were small (less than 3). However, the performance of the Hadoop system was better than the single PC system when its worker nodes were more than 3. When data size was increased, longer time was needed for data transmission in the Hadoop cloud system. Hence, to increase the performance of the Hadoop system, larger amounts of worker nodes should be used.

Table 3. Speedup ratios of Hadoop systems with various worker nodes over the single PC system

Data size (pieces)	1_node	2_node	3_node	4_node	5_node	6_node	7_node	8_node	9_node
1 M	0.58	0.83	1.38	1.45	1.71	1.72	1.86	2.08	2.14
800K	0.57	0.96	1.20	1.44	1.56	1.63	1.80	2.08	2.12
600K	0.57	0.89	1.10	1.48	1.56	1.84	1.96	2.03	2.13
400K	0.55	1.03	1.21	1.28	1.52	1.64	1.78	1.86	2.07
200 K	0.52	0.71	1.12	1.23	1.51	1.67	1.68	1.75	1.93
100 K	0.52	0.80	0.81	1.39	1.44	1.56	1.63	1.70	1.90

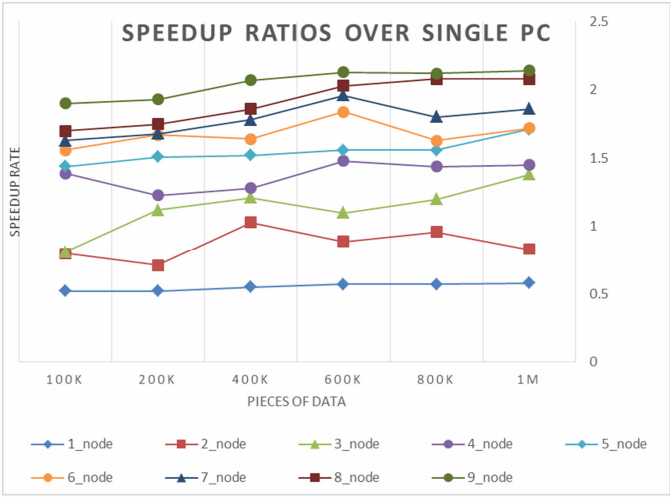

Figure 6. Speedup ratios for processing various amounts of data (in pieces) by Hadoop systems with various worker nodes over the single PC system

4.2 Experiment 2

In this experiment, we investigated the performance of the Hadoop system by varying the number of maps used in each node in MapReduce architecture [7]. There were 9 worker nodes and 400,000 pieces of news stored in the database. Execution times of Hadoop systems with various total numbers of maps (9 to 90 maps) in nodes are listed in Table 4 and shown in Figure 7.

Table 4. Execution times (in seconds) of the Hadoop system with various total numbers of maps used in nodes

9 maps (sec)	18 maps (sec)	27 maps (sec)	36 maps (sec)	45 maps (sec)	54 maps (sec)	63 maps (sec)	72 maps (sec)	81 maps (sec)	90 maps (sec)
2439	2046	1836	1784	1759	1753	1714	1839	1818	1821

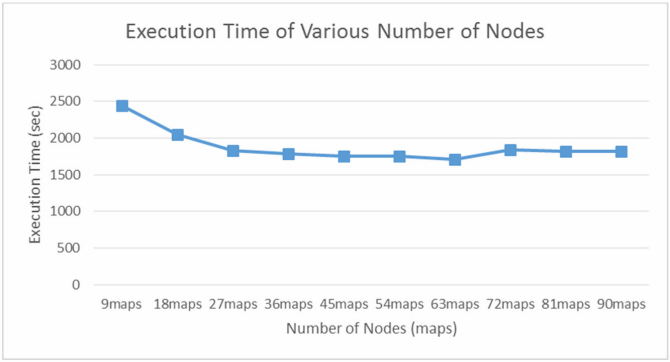

Figure 7. Execution times of Hadoop system with various total numbers of maps

Experimental results showed that the number of maps used in a node could be tuned to improve Hadoop system performance. The number of maps can be increased to improve performance. However, when the number of used maps became too large, it caused the database to queue up when the system detected

suspected personal information and queried the personal information database. According to experimental results, the optimal number of maps was determined by the speedup ratio (SR) in Equation (1):

$$SR = \frac{Execut\ time\ of\ 9\ maps}{Execut\ time\ of\ varied\ maps} \quad \textbf{(1)}$$

As shown in Table 5, the system performance was optimal when the total number of maps used in the 9 worker nodes was 63 (i.e., 7 maps per worker node), with a speedup percentage of 142%.

Table 5. Speedup ratios of performance with various total number of maps used over 9 maps

9 maps (sec)	18 maps (sec)	27 maps (sec)	36 maps (sec)	45 maps (sec)	54 maps (sec)	63 maps (sec)	72 maps (sec)	81 maps (sec)	90 maps (sec)
1.00	1.19	1.33	1.37	1.39	1.39	1.42	1.33	1.34	1.34

4.3 Discussion

With the aim to fulfill PIPA enacted in Taiwan since 2010, we have implemented an automata-based personal data protection system and investigated its efficiency. The experimental results show that the automata-based system implemented on a single PC can process data with a size less than 100K pieces within 18.6 minutes. However, since the execution time of the automata-based system grows linearly as the data size increases, the proposed Hadoop system is preferred to handle files of large volumes. In addition, performance of the proposed Hadoop system can be further enhanced by using a larger number of worker nodes and by choosing appropriate numbers of maps used in each worker node.

5 Conclusion

In this study, we first developed a personal data protection system using automata-based programming to analyze sentences and detect contents involving personal information. The implemented system consists of both client and server sides. On the client side, it can read documents in various formats and retrieve files transmitted via web pages and e-mails. The retrieved document is certificated by digital signatures and sent to the main server which adopts automata-based algorithm to analyze documents. When personal information is detected, it marks the location of found personal information and alerts users of personal information violations, thus greatly reducing the risk of personal information violation. In addition, the Hadoop cloud framework was further applied in the automata-based system to facilitate handling massive amounts of documents and accelerate the overall system performance. Experimental results showed that the proposed Hadoop system effectively accelerated execution time when larger number of worker nodes were used. In addition, the number of maps used in each worker node could be tuned to optimize system performance. In the future, to further enhance the performance of the cloud-based personal data protection system, we plan to use more physical machines and expand worker nodes in the Hadoop cloud system. In addition, we also research on combining machine learning approaches with our system structure to develop the personal data protection system.

Acknowledgments

This research was partly supported by Ministry of Science and Technology, Taiwan, under grant number: MOST 107-2221-E-029-005-MY3.

References

[1] D. Wang, Y. Wu, W. Zhao, L. Fu, A Model of Privacy Preserving in Dynamic Set-valued Data Re-publication, *Journal of Internet Technology*, Vol. 20, No. 1, pp. 147-156, January, 2019.

[2] H. Long, L. Zhang, J. Wang, S. Zhang, An Incentive Mechanism with Privacy Protection and Quality Evaluation in Mobile Crowd Computing, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 30, No. 3, pp.187-198, January, 2019, DOI: 10.1504/IJAHUC.2019.10019942.

[3] Y. Liu, Y. Zhou, Y. Tian, M. Liu, Y. Zheng, Secure and Lightweight Remote Medical System, *Journal of Internet Technology*, Vol. 20, No. 1, pp. 177-185, January, 2019.

[4] National Development Council, Personal Information Protection Act, https://law.moj.gov.tw/ENG/LawClass/LawParaDeatil.aspx?pcode=I0050021&bp=6.

[5] Apache Software Foundation, Apache Hadoop, http://hadoop.apache.org.

[6] J. Dean, S. Ghemawat, *MapReduce: Simplified Data Processing on Large Clusters*, Google Inc., 2004.

[7] S. Perera, T. Gunarathne, *Hadoop MapReduce Cookbook*, Packt Publishing, 2013.

[8] J. Shafer, S. Rixner, A. L. Cox, The Hadoop Distributed Filesystem: Balancing Portability and Performance, *2010 IEEE International Symposium on Performance Analysis of Systems & Software*, White Plains, NY, USA, 2010, pp. 122-133.

[9] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The Hadoop

Distributed File System, *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies*, Lake Tahoe, Nevada, USA, 2010, pp. 1-10.

[10] F. Yu, T. Bultan, M. Cova, O. H. Ibarra, Symbolic String Verification: An Automata-based Approach, *The 15th International Workshop on Model Checking Software*, Los Angeles, CA, USA, 2008, pp. 306-324.

[11] Apache Software Foundation, Apache POI, https://poi.apache.org.

[12] J. Hedley, Jsoup: Java HTML Parse, https://jsoup.org.

[13] Y. Song, Z. Liao, Y. Liang, A Trusted Authentication Model for Remote Users under Cloud Architecture, *International Journal of Internet Protocol Technology*, Vol. 11, No. 2, pp. 110-117, June, 2018.

[14] Y. Wang, Privacy-Preserving Data Storage in Cloud Using Array BP-XOR Codes, *IEEE Transactions on Cloud Computing*, Vol. 3, No. 4, pp. 425-435, October-December, 2015.

[15] M. Al-Ruithe, E. Benkhelifa, Y. Jararweh, C. Ghedira, Addressing Data Governance in Cloud Storage: Survey, Techniques and Trends, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 1763-1775, November, 2018.

[16] Apache Software Foundation, *Apache Hadoop YARN*, http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html

Biographies

Jung-Chun Liu received his M.S. and Ph.D. degrees from the Department of Electrical and Computer Engineering at the University of Texas at Austin, in 1996 and 2004, respectively. He is currently an associate professor in the Department of Computer Science at Tunghai University, Taiwan. His research interests include cloud computing, embedded systems, big data, network security, and artificial intelligence.

Chu-Hsing Lin received his Ph.D. degree in computer science from National Tsinghua University, Taiwan, in 1991. He is currently a professor at the Department of Computer Science, Tunghai University. Professor Lin has ever been the director of computer center, chairman of the department, and the library director. His current research interests include information security, cryptography, machine learning, and data science.

Ken-Yu Lee received his B.S. and M.S. degrees in computer science from Tunghai University, Taiwan, in 2014 and 2017, respectively. He was awarded the Gold Penguin Award by the Ministry of Economy, Taiwan. His two papers won the first prize and the best paper awards in the IEEE ICASI 2017 international conference. His current research interests include machine learning, mobile application and Web service.