TEST Engineering & Management

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

Privacy and Big Data Protection: Comparisons between Data Encryption Methods

Lau Boon Leong, School of Computer Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia.
Ooi Tiat Han School of Computer Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia.
Law Yuan Hong, School of Computer Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia.
Pantea Keikhosrokiani, School of Computer Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia.
Rosni Abdullah, School of Computer Sciences, Universiti Sains Malaysia, 11800 Minden, Penang, Malaysia. National Advanced IPv6 Centre of Excellence (Nav6), Universiti Sains Malaysia 11800 Minden, Penang, Malaysia

Article Info
Volume 82
Page Number: 11923 - 11930
Publication Issue:
January-February 2020

Article History
Article Received*: 18 May 2019*
Revised*: 14 July 2019*
Accepted*: 22 December 2019*
Publication*: 21 February 2020*

Abstract

Abstract: Big data revolutionized many industries. Big data security became a challenge that requires constant updates and modifications. For this reason, this study implements encryption and decryption models in data storage phase to protect big data from data breaches. Three encryption algorithms are tested by applying specific dataset. The algorithms are implemented to generate the performance metric. The encryption and decryption times are further analyzed and discussed. Despite obtaining the run time of the encryption methods, it is inferred that these three encryption methods are served for different purposes and the comparison is minimal. Generally, the runtime of the three algorithms increase linearly along the data size. Identity based encryption provides lower computational cost with only a certain level of confidentiality, while attribute based encryption provides higher security level by increasing computational cost without limits. Homomorphic encryption is inferred as the most secure encryption method by assuring no deciphering during all computations.

Keywords: *Big Data, Data Storage, Encryption, Security.*

I. INTRODUCTION

In the era where the data is growing, at an unprecedented scale than our ability to analyze it, the advent of big data is here. It is predicted that the size of our digital world will at least be double every two years and it will be a 50-fold expansion from 2010 to 2020 [1]. Big data can be defined as but not limited to structured, unstructured, geographic, real-time media, natural language, time series, network or linked data [2]. Generally, big data requires distributed systems to fulfill the high computational power and large storage prerequisites. Hence, understanding the design of existing algorithms to safeguard the big data becomes a crucial role for the organizations. The international organizations like Facebook, Google and Twitter contribute to produce more than 500 terabytes of data for their daily operation [3]. Big data are often used to analyze the user's behavioral patterns especially on shopping sites.

Security is required for big data protection in different phases including data generation phase, data storage phase, and data processing phase. This study focus on securing data storage phase. This study involves implementing encryption and decryption models to protect big data from data breaches. There are different encryption schemes for securing data storage phase including Attribute Based Encryption (ABE), Identity Based Encryption (IBE), Homomorphic encryption, Storage Path Encryption, Hybrid Clouds among which the first three methods have less limitations [5]. Therefore, the models that are used for encrypting and decrypting big data in this study consists of Attribute Based Encryption (ABE), Identity Based Encryption (IBE) and Homomorphic

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

Encryption that are used in some researches [4-6]. The ABE assures point to point big data privacy in cloud systems. By using ABE, only users whose attributes satisfy the policies defined by the data owner can decrypt the data [5]. The IBE utilizes public-key encryption where the public key of user provides some unique details about his or her own identity. The unique information could be e-mail addresses, Internet Protocol addresses, phone numbers, and dates [4]. The Homomorphic Encryption is an encryption strategy that produces a reversible plaintext and ciphertext. It implies that the encrypted results match exactly with its original plaintext after decryption and vice versa [6]. The study of encryption and decryption models allows the organizations to better maintain integrity of data after encryption and protect sensitive data of users as well as their privacy. The scope of this study is limited to techniques of encrypting and decrypting big data in various types via available algorithms built using certain models and the data sets are from several web applications.

II. RELATED WORKS

Data lifecycle management determines what kind of data to be stored and discarded throughout the analytical process [5]. In this section, the discussion begins with the understanding of the characteristics of security mechanisms such as methods or algorithms.

There is an explosive growth of data and all of these data are required to be stored, analyzed, categorized and utilized. Huge amounts of clients' private and secret data along with metadata will be stored in data centers and require protection during processing and transmission [7]. Hence, data storage requires a certain level of management. Various storage systems had emerged to meet the demands of all these massive data and one of the most significant and successful storage systems would be cloud storage [5]. Outsourcing to clouds is one of the most common ways in order to secure big data storage [8],[9],[10], in which data owners encrypt their data using cryptography algorithms and store them on clouds. When data are stored on cloud, there are 3 main dimensions, confidentiality, integrity and availability [5]. Any breach of confidentiality or integrity would mean that the privacy of data owner is in risk.

Big Data's potential is being expanded to the fullest with the help of cloud storage, and this requires new ways to protect the data instead of any other traditional methods. Security functions have to cover more than one aspects, such as hardware, operating systems, etc. All these security concerns require a certain applicable mechanism so that the privacy of the Big Data could be secured. Table I shows the few encryption methods that are able to be integrated into cloud storage along with the comparison of each encryption method.

Table I. Comparison of data protection in data storage phase.

Encryption Scheme	Features	Limitations
Attribute Based Encryption	- Data access control is based on predefined users' attributes. - Allow more flexibility in specifying different users' access right with Fine- grained Access Control, a feature introduced by Key-Policy Attribute Based Encryption (KP-ABE).	- Data to be updated must undergo the process of downloading, decryption, encryption and reuploading. Computational overhead would be very high for Big Data.
Identity Based Encryption	- Data access control is based on the identity of a user. - Complete access over all resources. - Relatively good security level, as there is preservation towards the sender and recipient's identity.	- The risk of data disclosure is very high if the centralized server had been compromised. - Data to be processed must be downloaded and decrypted. - Considering IBE is being used in a larger environment, computational process such as

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

		authentication, decryption, etc. will be time consuming. - High computational overhead.
Homomorphic Encryption	- Very secure. - Computations are performed on ciphertext (encrypted data).	- May have high complexity when performing computations on ciphertext.

a. Attribute Based Encryption

Attribute Based Encryption (ABE) is an encryption technique which ensures end to end big data privacy in cloud storage system. It is able to provide flexible access control and data confidentiality functionalities simultaneously [11]. By using ABE, data can only be decrypted by users whose attributes satisfy the policies defined by the data owner. However, this might be a problem when we are dealing with big data as the data is always changing and the policies might also require constant updates. The policy updating is a very challenging task in attribute-based access control systems. Once the data is outsourced to cloud storage, local copy will not be kept by data owners in their system. This raise a problem when there is a need to update the data's policies stored in cloud. The data owner would have to transfer the data back to local system, re-encrypt the data under new policy and reupload it back to cloud. This process requires very high communication overhead and computational cost and it is not practical for Big Data [5]. ABE is designed for "one-to-many" property, in which a single key can decrypt various ciphertexts encrypted with different attributes. In addition, two different keys can decrypt the same ciphertext if and only if the key and the ciphertext satisfy the decryption condition which is referred to as policy [12].

b. Identity Based Encryption

Identity Based Encryption (IBE) is derived from Public Key Encryption (PKE), an encryption scheme that uses a pair of keys; public key and private key. In PKE, the public key can be accessed by anybody to encrypt a certain message or data while only to be received and decrypted by the desired recipient with the private key [5]. IBE schemes use any public information to create the public key such as e-mail addresses, Internet Protocol addresses, phone numbers, and dates [13]. The sender will create an encrypted message along with the required parameters which includes identity (e.g.: email). The parameters will then be generated into a public key and send to the server. When the recipient receives the encrypted message from the server, he or she could then use his/her identity details (e.g.: email) to decrypt the message upon authentication. With IBE, the anonymity of sender and receiver can be preserved to some extent, such as less vulnerable to spam messages. The issue of looking up a user's public key no longer exists as IBE public keys are calculated by using user's public information [new-4]. However, for IBE to work flawlessly, it requires a centralised server or a third-party key generator. All message or data sent will have to go through the central server before it was further sent to all receivers. The weakness of having a centralised server is that if the server had been compromised, the risk of disclosure is very high as all messages and data that had been encrypted by using the PKE are also compromised [5].

c. Homomorphic Encryption

Homomorphic encryption is one of the proposed way to deal with the vulnerability of public cloud against privacy breaches. The reason why public cloud is so vulnerable is because cloud users may share the same physical space (multi-tenancy) and the chances of this scenario leading to data leakage is very high. Homomorphic encryption is a form of encryption which allows computational process to be performed on ciphertext (encrypted data) without compromising the plaintext (original data). The biggest advantage of using homomorphic encryption is that full privacy is provided to the data owners. But at the same time, complexity level might increase during certain computational process as the ciphertext is used instead of plaintext [5]. Homomorphic encryption can be classified into various schemes which are Partially Homomorphic Encryption (PHE), Somewhat

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE). PHE only allows some operations to be performed on encrypted data. For example, addition and multiplication are the two given operations, only one of them can be performed on the encrypted data. SHE supports more than one operation to be performed on encrypted data but not all operations can be applied to all types of data. FHE supports any number of operations on any encrypted data. Even though FHE supports operations for any encrypted data, it is less efficient than PHE and SHE because of the computation overhead [15].

III. METHODOLOGY

There are various encryptions methods for data privacy protection in data storage phase. Three methods are chosen in these studies which are Attribute Based Encryption, Identity Based Encryption and Homomorphic Encryption. This research is conducted by the application of algorithms and online source code provided by researchers. By implementing the algorithms, a deeper understanding of the encryption method and its overall performance could be obtained. Three source code libraries had been referred during the research which is Jpair [16], cpabe [17] and HElib [18]. Jpair and cpabe library supports Java programming language and Jpair is implemented for identity-based encryption, while cpabe is implemented for attribute- based encryption. As there are many branches and derivations of attribute-based encryption scheme, ciphertext policy attribute-based encryption is used in this research. For homomorphic encryption, HElib is implemented with C++ programming language. By implementing the algorithms referred, the three encryption methods are analyzed from the factor of time as run time is decent performance metric. The simulation process is done on laptop with specifications of Windows 64-bit, processor i5-6200U and CPU 2.30GHz with 4GB of DDR4 2133MHz RAM. Note that the simulation device plays a big role of attaining the results especially performance benchmark such as time. Newer version of processors and random-access memory (RAM) used by researchers will guarantee better performance of the encryption methods. The outputs of the encryption methods are analyzed and compared after obtaining the results.

IV. EXPERIMENTS AND RESULTS

a. Attribute Based Encryption

The performance of attribute-based encryption could be determined by the measurements of private key generation time, encryption time and decryption time. Ciphertext policy attribute-based encryption is analyzed and researched by referring to cpabe toolkit. cpabe-keygen generates a private key with a given set of attributes and the run time increases as the number of attributes increases.

For encryption time, the run time depends on the complexity of the encryption policy. For example, the complexity could vary according to the job position in a company. A senior executive position will have a higher complexity compared to a business staff. Nevertheless, the encryption run time increases as the complexity increases (Fig. 1).

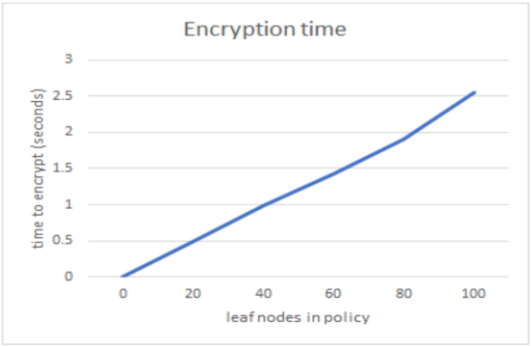

Fig. 1. Encryption Time for Attribute Based Encryption.

Decryption time also contributes to the overall performance of this encryption scheme. However, compare to encryption time, decryption time is slightly lower but still increasing as the complexity increases (Fig. 2).

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

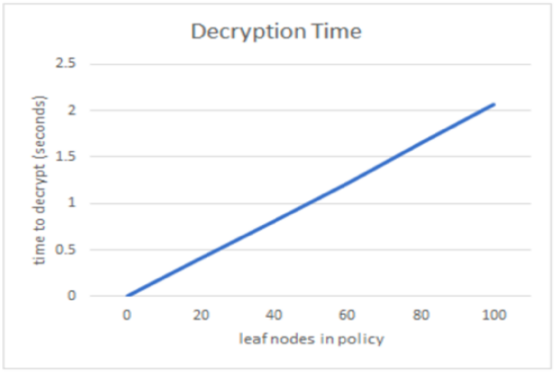

Fig. 2. Decryption Time for Attribute Based Encryption.

b. Identity Based Encryption

The results of the Identity-based encryption are discussed in term of size of identity value, length of public and private keys and also time required to encrypt and decrypt the message. When the user identity value is getting bigger in size, the longer it is required to generate respective public and private keys.

Fig. 3 and Fig. 4 indicate the time taken to encrypt and decrypt the message to be sent vary as the message length increases. It is mainly due to fact that it requires longer time to convert the message to ciphertext. However, the decryption time is far shorter than encryption time as the length of private key used to decrypt the message is smaller in size than that of public key.

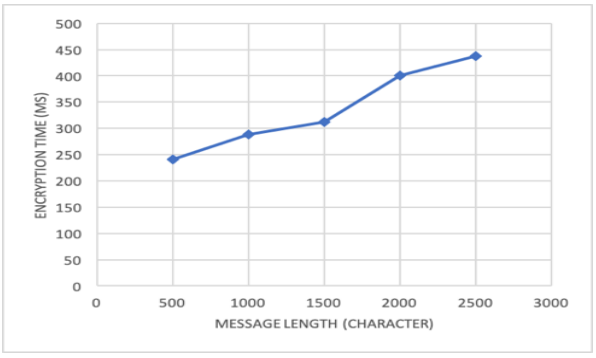

Fig. 3. Encryption Time for Identity Based Encryption.

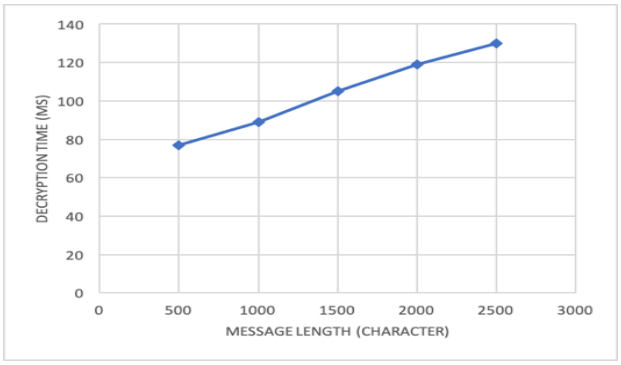

Fig. 4. Decryption Time for Identity Based Encryption.

c. Homomorphic Encryption

The result of Homomorphic encryption technique simulation is discussed in term of encryption time vs data size and decryption time vs data size. HElib is implemented and analyzed with C++ programming language. As shown in Fig. 5, the data size is the manipulating variable while the encryption time is the responding variable. Observation that can be deduced is that the encryption time increases as the data size increases.

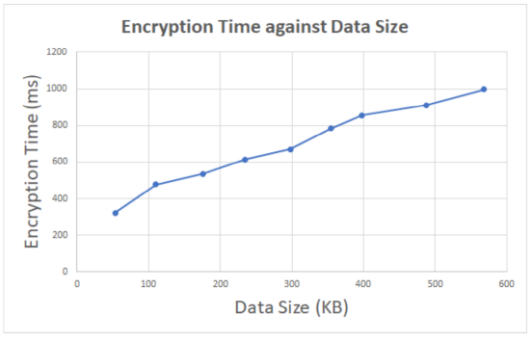

Fig. 5. Encryption Time for Homomorphic Encryption.

Based on Fig. 6, the data size is the manipulating variable while the decryption time is the responding variable. Observation that can be deduced is that the decryption time increases as the data size increases. The decryption time is higher than the encryption time with the same data size.

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

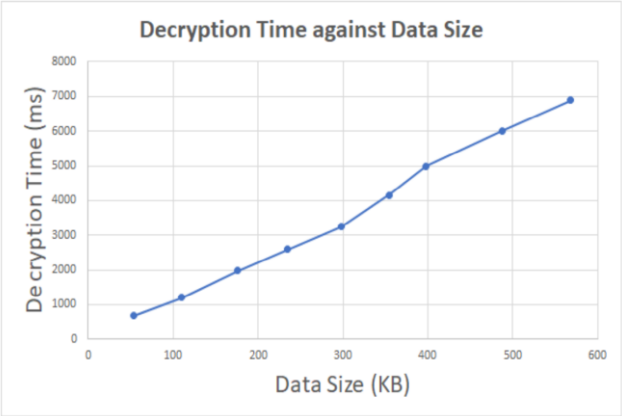

Fig. 6. Decryption Time for Homomorphic Encryption.

V. DISCUSSION AND CONCLUSION

Typical public-key cryptography requires recipient to have both public and private keys whereas the sender has known the recipient's public key. One of the key advantages for identity-based encryption is that it uses a simple identifier such as email address to generate a public key that can be used for encrypting and decrypting electronic message. For example, the sender can generate a public key using recipient's email address to encrypt the message to be sent to recipient. It greatly reduces the complexity of encryption process for both users and admins compared to typical public-key cryptography. The disadvantages are that the identity-based encryption requires a centralized server and a secure channel between sender and recipient and the server for transmitting the private key. Attribute based encryption (ABE) is primarily used for data sharing. For example, it can be used for sending messages to specific receivers. Compare to identity-based encryption which the whole encryption scheme is based on the authentication of an identity (e.g. email), ABE encrypts data with a series of attributes into a private key. For our research, we implemented ciphertext policy ABE, a derivation algorithm from basic ABE. This algorithm includes a policy rules and could increase the security level as more authentication is involved during the encryption. The degree of security could also be further enhanced by involving more attributes with the cost of computational time. By referring to Fig. 1 and Fig. 2 generally as the number of attributes and complexity increases, the run time increases too. Hence, the advantage of ABE scheme that allows stronger security invokes the weakness of having higher overhead in return. The similarity between identity-based encryption and ABE is that both of these encryptions relies on a trusted distributor or channel for transmitting the private key. Homomorphic encryption is commonly used for secure outsourced computation because computation can be performed without exposing the unencrypted data. Although the decryption time for homomorphic longer than the encryption time, homomorphic encryption is widely used in computation because computation can be done using the encrypted data. The result of the computation can be decrypted and hence privacy of the data can be protected. The disadvantage of homomorphic encryption is that the performance of computation is lower because the computation of encrypted data is more complex than original data.

The full potential and possibilities of cloud computing has yet to be discovered and determined. Large scale cloud-based transformation is arriving at the doorstep of not only technology field, but business, manufacturing industry, etc. as well. All these revolutions will always lead a challenge, data security and privacy protection. Big data protection is crucial in preventing data leakage, unauthorized access, etc. Three different encryption schemes had been tested and evaluated by implementing the algorithms from researchers. Despite having certain similarities in the ideology of implementing the encryption, we can conclude that each of them serves different purposes. Identity based encryption has a relatively lower computational cost but still able to ensures high level of confidentiality. All private keys are easy to be managed as each of them only contains only the identity of the recipient. No additional information means that the chances of unnecessary access to sensitive information are greatly reduced. The major drawback of identity-based encryption is

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

that it is highly dependent on the private key generator just like attribute-based encryption. Despite having the drawback, attribute-based encryption still has its own uniqueness. As the generation of private key in attribute-based encryption involves attributes of recipient, the sensitivity of the encryption is increased and thus higher security level is allowed. Homomorphic encryption method is arguably the most secure method among all three encryption methods that we looked into. This is mainly due to the capability of homomorphic encryption that allows computation on the ciphertext (encrypted data) which could not be done by basic identity-based encryption and attribute- based encryption. This capability allows homomorphic encryption to be implemented by many services that involves monetary actions such as transaction, etc. Ciphertext could be run by any third party and this includes the untrusted ones without revealing sensitive information, inputs and the internal state of the ciphertext. In terms of performance, homomorphic encryption is typically slower than non-homomorphic encryption due to the fact that computation is always faster if performed in plaintext compared to ciphertext.

ACKNOWLEDGMENT

The authors are thankful to School of Computer Sciences, Universiti Sains Malaysia for funding this paper and for the unlimited supports.

VI. REFERENCES

[1] Team, E. (2018). The Exponential Growth of Data. [online] insideBIGDATA. Available at: https://insidebigdata.com/2017/02/16/the-exponential-growth-of-data/ [Accessed 2 Nov. 2018].

[2] C. Xie, J. Gao, and C. Tao, "Big Data Validation Case Study," in 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), 2017, pp. 281-286: IEEE.

[3] A. Kadadi, R. Agrawal, C. Nyamful, and R. Atiq, "Challenges of data integration and interoperability in big data," in 2014 IEEE International Conference on Big Data (Big Data), 2014, pp. 38-40: IEEE.

[4] N. McCullagh, "Securing e-mail with identity-based encryption," IT professional, vol. 7, no. 3, pp. 64, 61-63, 2005.

[5] A. Mehmood, I. Natgunanathan, Y. Xiang, G. Hua, and S. Guo, "Protection of big data privacy," IEEE access, vol. 4, pp. 1821- 1834, 2016.

[6] R. Hayward and C.-C. Chiang, "Parallelizing fully homomorphic encryption," in 2014 International Symposium on Computer, Consumer and Control (IS3C), 2014, pp. 721-724: IEEE.

[7] C. Hongbing, R. Chunming, H. Kai, W. Weihong, and L. Yanyan, "Secure big data storage and sharing scheme for cloud tenants," China Communications, vol. 12, no. 6, pp. 106-115, 2015.

[8] L. Wei, "Security and privacy for storage and computation in cloud computing", Inf. Sci., vol. 258, pp. 371-386, 2014.

[9] H. Cheng, C. Rong, K. Hwang, W. Wang, Y. Li, "Secure big data storage and sharing scheme for cloud tenants", China Communications., vol. 12, no. 6, pp. 106-115, 2015.

[10] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid", IEEE Trans. Cloud Computing., vol. 3, no. 2, pp. 233-244, Apr.–Jun. 2015.

[11] A. Ge, J. Zhang, R. Zhang, C. Ma, and Z. Zhang, "Security analysis of a privacy- preserving decentralized key-policy attribute-based encryption scheme," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 11, pp. 2319-2321, Nov. 2013.

[12] W. Qiuxin, "A generic construction of ciphertext-policy attribute-based encryption supporting attribute revocation," China communications, vol. 11, no. 13, pp. 93-100, 2014.

[13] N. McCullagh, "Securing e-mail with identity-based encryption," IT professional, no. 3, pp. 64, 61-63, 2005.

[14] L. Martin, "Identity-based encryption comes of age," Computer, vol. 41, no. 8, 2008.

[15] M. Mohan, M. K. Devi, and V. J. Prakash, "Homomorphic encryption-state of the art," in 2017 International Conference on Intelligent Computing and Control (I2C2), 2017, pp. 1-6: IEEE.

[16] "Jpair," SourceForge. [Online]. Available: https://sourceforge.net/projects/jpair/. [Accessed: 20-Dec-2018].

[17] J. Wang, "cpabe," cpabe by junwei-wang. [Online]. Available: https://junwei.co/cpabe/. [Accessed: 20-Dec-2018].

[18] S. Halevi and V. Shoup, "Faster Homomorphic Linear Transformations in HElib," Lecture Notes in Computer Science Advances in Cryptology – CRYPTO 2018, pp. 93–120, 2018.

January - February 2020
ISSN: 0193 - 4120 Page No. 11923 - 11930

AUTHORS PROFILE

Lau Boon Leong is a final year student at School of Computer Sciences, Universiti Sains Malaysia (USM). His specialization is Distributed Systems and Security.

Ooi Tiat Han is a final year student at School of Computer Sciences, Universiti Sains Malaysia (USM). His specialization is Information System Engineering.

Law Yuan Hong is a final year student at School of Computer Sciences, Universiti Sains Malaysia (USM). His specialization is Software Engineering.

Pantea Keikhosrokiani Pantea Keikhosrokiani is a Senior Lecturer at the School of Computer Sciences, Universiti Sains Malaysia (USM; Penang, Malaysia). She was a teaching fellow at the National Advanced IPv6 Centre of Excellence (Nav6), USM. She has received her PhD in Service System Engineering, Information System and her master degree in Information Technology from the School of Computer Sciences, USM. She has been graduated in Bachelor of Science in Electrical Engineering Electronics. In her PhD, she has particularly focused on developing and integrating mobile healthcare information systems to provide location-based healthcare monitoring services for patients affected by arrhythmia and hypertension. In addition, she has worked on location-based mobile commerce information system for her master degree. Her articles have been published in distinguished edited books and journals including Elsevier (Telematics & Informatics), Springer (Cognition, Technology, & Work), Taylors and Francis and IGI global, and have been indexed by ISI, Scopus and PubMed. She reviewed papers for conferences and distinguished journals related to information systems, health and medical informatics, business informatics, business informatics, etc. She also has worked as a teaching assistant at the School of Computer Sciences, USM, as part of her duty for the prestigious USM Fellowship, which was granted to her for 3 years. Her areas of interest for research and teaching are Information Systems Development, Database Systems, Health and Medical Informatics, Business Informatics, Location-Based Mobile Applications, Big Data, and Technopreneurship.

Professor Dr Rosni Abdullah is a Professor in Parallel Computing at the School of Computer Sciences, Universiti Sains Malaysia (USM) and is one of the national pioneers in this field. She is currently dean of School of Computer Sciences, USM and Director of the National Advanced IPv6 Centre (Nav6) which is a centre of research excellence in USM that focus on future internet research aligned towards the Internet of Things (IoT). She obtained her PhD in April 1997 from Loughborough University, United Kingdom specializing in the area of Parallel Numerical Algorithms. Both her Bachelors degree and Masters degree in Computer Science were obtained from Western Michigan University, Kalamazoo, Michigan, U.S.A. in 1984 and 1986 respectively.

Rosni has served as Dean of the School of Computer Sciences at Universiti Sains Malaysia (USM) from 2004 to 2012, after having served as its Deputy Dean (Postgraduate and Research) since 1999. She is also the Head of the Parallel and Distributed Processing Research Group at the School since its inception in 1994.

Her research areas include Parallel and Distributed computing, Parallel numerical algorithms and Computational support for applications of bioinformatics. She has led more than 20 research grants including two European Union grants and two Intel grants, and has published more than 100 papers in journals and conference proceedings.