International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 6, December 2016, pp. 2979~2988
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i6.11573

Finding the Optimal Value for Threshold Cryptography on Cloud Computing

Weena Janratchakool, Sirapat Boonkrong, Sucha Smanchat
Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, Thailand

Article Info

Article history:

Received Jun 17, 2016
Revised Nov 10, 2016
Accepted Nov 25, 2016

Keyword:

Threshold cryptography
Distribution time
Reconstruction time
Cloud simulation

ABSTRACT

The objective of using threshold cryptography on cloud environment is to protect the keys, which are the most important elements in cryptographic systems. Threshold cryptography works by dividing the private key to a number of shares, according to the number of virtual machines, then distributing them each share to each virtual machine. In order to generate the key back, not all the shares are needed. Howerver, the problem is that there has been no research attemping to find a suitable threshold value for key reconstruction. Therefore, this paper presented a guildline designed and implemented that can assist to choose such value. The experiment was setup using CloudSim to simulate cloud environment and collecting time taken in key distribution and key reconstruction process to achieve the optimal threshold value.

Corresponding Author:

Weena Janratchakool,
Faculty of Information Technology,
King Mongkut's University of Technology North Bangkok,
Bangkok-10800, Thailand.
Email: yweena@icloud.com

1. INTRODUCTION

Cloud computing technology has been growing rapidly due to its essential characteristics which consist of on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service [1]. More enterprises consider moving data to cloud [2]. Therefore, a large amount of data, which usually consists of sensitive or important data, is also sent into the cloud. Without doubt, the attacks are also increasing accordingly. For this reason, the challenges with data protection are the serious security issue. Many techniques for protecting data such as cryptography have been applied in the cloud to reduce the risk of data being compromised.

Cryptography is a mathematical based tool for providing security over the network. Its process is associated with converting plaintext into cipher text known as encryption and then back again called decryption. There are two different types of cryptography. The first is symmetric cryptography, which using the same key in both encryption and decryption process. The second is asymmetric cryptography, which using public key for encryption and using private key for description. This is the type, specifically RSA [3] that all experiments in this paper will be addressed.

Due to the most important element for cryptography process is the key according to Kerkhoff's rule [4], which states that, "a cryptosystem should be secure even if everything about the system is publicly known, except the key." This means that the protection of the key is also the fundamental of the overall security of data protection.

As basic concept, cloud computing is totally internet-based technology providing all of resources as services delivered in the form of virtual machines (VMs). Therefore, most data is stored in the VM that also involving the key. VMs, of course, have become more attractive attack targets for adversaries to steal the key and expose secret information such as side-channel attack [5-7] and software-based attack [8]. Athough there

is already Key Management Authority (KMA), which provides the keys to encrypt and decrypt data before storing in the database over the access policy [9], it makes perhaps more vulnerability. If the Key Management Authority (KMA) is compromised so, the overall of cloud system must be attacked.

In order to offer more security on the cloud, the threshold cryptography is introduced to protect the key being compromised. The basic concept of threshold cryptography is that the key is divided into *n* shares before being distributed to the involved entities. In key reconstruction, only *k* shares (known as threshold value) able to combine not all the shares are needed.

The threshold cryptography scheme seems to be an advanced step to providing security for the key. This is because the adversary will need to attack *k* entities in order to obtain the recombined key that make it more difficult for the attacker.

Although, there are some researchs using the threshold cryptography to provide security for key stored on cloud [10-14], there are no researches attempting to find a suitable threshold for any number of shares distributed. The objective of this paper is, therefore to design and implement a method that is able to assist in finding such values. In addition, time for key distribution and key reconstruction are collected to analyze what desireable threshold value should be by using CloudSim toolkit.

The rest of the paper is organized as follows. Section 2 presents brief background knowledge on threshold cryptography and cloud computing involving the cloud simulation tool known as CloudSim. The experimental designs are explained in Section 3. The results of the experiment and discussion are given in Section 4. Finally, Section 5 concludes the paper.

2. BACKGROUND KNOWLEDGE

This section describes the fundamental architectrue of cloud computing simulation as CloudSim and the algorithms of threshold cryptography, (*n, k*) - Shamir's secret sharing scheme, based on RSA for finding suitable parameters.

2.1. Overview of Threshold Cryptography

Threshold cryptography is a secret-key sharing scheme for public key cryptography, denoted as (*n, k*)-threshold, introduced by Shamir [3]. The algorithm works by dividing the private key, generated by RSA keypairs generator, into sub *n* keys or shares according to number of entities, which are number of created virtual machines (VMs) on single host. The key recombination operates by collecting the partial of *k* keys from *n* shares distributed among them able to be reconstructed into form of original private key. Otherwise, only a few subkeys or shares can be collected to generate the private key back.

Accordingly, the threshold cryptography scheme is generally distinguished into two phases: shared-key distribution and shared-key reconstruction that are the significant factors in this paper.

2.1.1. Shared-key distribution phase

At the beginning, a private key is generated using any asymmetric cryptographic algorithm, in here is RSA. The private key is then being spit into *n* shares according to entire entities, whereas VMs, in the system based on polynomial interpolation. The total amount of *n* subkey shares is then distributed to each entity. Therefore, the process of this phase can be summarized as follows.

Step 1: The private key is generated by RSA algorithm, which is one mechanism of asymmetric cryptography.

Step 2: The private key is devided into subkeys in form of (*n, k*)-threshold scheme based on a random polynomial of degree *k-1*, whereas the coefficient a_0 is private key shown in (1).

$$f(x) = a_o + a_1 x + \cdots + a_{k-1} x^{k-1} \quad (1)$$

Step 3: The new sub secret key shares are distributed to each entity.

2.1.2. Shared-key Reconstruction Phase

When the private key needs to be recombined, number of *k* subkey shares from number of *n* subkey shares would be collected. In other words, only *k* key shares are needed to create key back. Here, *k* is known as a threshold value.

The key reconstruction process is operated by applying the Lagrange Interpolation shown in (2).

$$f(x) = \sum_{i=1}^{k} y_i \left(\prod_{1 \le j \le k, j \ne i} \frac{x - x_j}{x_i - x_j} \right) \quad (2)$$

2.2. Overview of CloudSim

CloudSim [15] is one of cloud simulation toolkit that allows for virtualized environments, which enable model, experiment and test performance of applications and services before launch on real-world cloud. CloudSim is library tool for simulation cloud scenario written in Java language. There are many researches using CloudSim toolkit such as to study CPU allocation and scheduling algorithms [16] and to simulate and optimize the execution time and average waiting time in cloud environment [17]. In CloudSim, there are, of cause, many different classes that support cloud environment simulation. Therefore, more details of some classes concerned are also necessarily explained consisting of Cloudlet, VM, Host, Datacenter and DatacenterBroker.

Cloudlet class is represented application tasks or services. Each cloudlet object encapsulated the number of instructions, overhead of data transfer, model of workload generation and identification of VM, which is assigned on. VM is a class, represented virtual machine, provided characteristics of each virtual machine including processing power, RAM, network and bandwidth, storage size and host, which it's locating. Host class is modeled the physical resources such as datacenters and storage servers. Its characteristics consist of lists and types of processing cores (PE), amount of storage, RAM and policies for allocating VM. Moreover, host is also directly associated to Datacenter.

Datacenter class models the core infrastructure, which encapsulates lists of hosts. It deals with scheduling of each VM to place on any host. DatacenterBroker is modeled to response for mediating between users and services.

2.3. Related Work

Data protection and key management are two of most important concerns for cloud security. To overcome some threats such as side-channel attack [5-7], therefore, threshold cryptography is one of solutions conducted to provide more security on cloud. There have some researches already been done for using threshold cryptography.

Cloudstash [10] applies the secret-sharing scheme directly on the file that spit into multiple shares of secret and then distributed them into multiple clouds simultaneously where shares are required to reconstruct the file. The InterCloud [11] system works by using symmetric encryption on the data and spitting the key into shares using secret sharing scheme. Before distributing to the cloud, the each share-key is attached to the pices of data as metadata. The CloudSeal [12], an end-to-end content confidentiality protection scheme for large-scale content storage, integrates symmetric encryption, threshold secret sharing and proxy based re-encryption scheme to protect content and mange the user access.

As for key protection aspect, a multilevel threshold secret sharing scheme [13] is conducted by duplicating the secret key and then distributing into multiple resource providers to ensure availability. Moreover, key-insulated symmetric cryptography [8] is presented to against the compromise of keys from software-based attack such as malware. In authentication, threshold cryptography is conducted to integrate with Kerberos [14] protocol to provide more security and to increase availability of key.

Although, there are some researchs using the threshold cryptography to provide security on cloud, there are no researches attempting to find a suitable threshold for any number of shares distributed.

3. RESEARCH METHOD

This section explains experimental designs to find a suitable threshold value, *k*, for any number of *n* shares, in the cloud environment using CloudSim. In order to simulate cloud environment, five new classes, designed for dealing with especially the keys, consisting of KeyDistribution, KeyGeneration, Key Reconstruction, Key Shared and Prime RSA Number are added in Cloud Sim framework. Key Generation, Key Distribution and Key Reconstruction classes are built to handler with generating the private key based on RSA algorithm, distributing the keys to virtual machine and collecting keys to reconstruct the key respectively. Key Shared class is designed for storing behavior of each key share.

Figure 1 shows the process of key generation and distribution on the cloud simulation (Cloud Sim) that consists of KeyGeneration class, KeyShares class and Keydistribution class. The process starts with the virtual machines (VMs) requesting from cloud user to DatacenterBroker class that provides the data c enter (s) and host (s) by sending the request to Datacenter and Host class repectively. When the data center and host have been already created, the virtual machines (VMs) will be requested from user via the DatacenterBroker class. In addition, DatacenterBroker class is entended to take responsibility for assigning keys to each virtual machine.

In the KeyGeneration class, the 2048 bit-keysize of private key is generated by using the RSA algorithm. Then the private key is sent to KeyShares class to divide the key into subkeys in form of (n, k)-threshold scheme based on a random polynomial of degree *k-1* according to equation in (1).

The (n-k)-threshold scheme of subkeys is stored in KeyShares class that is designed for storing the behavior of each key shares. Then, they are distributed to every virtual machine (VMs) by Keydistribution class.

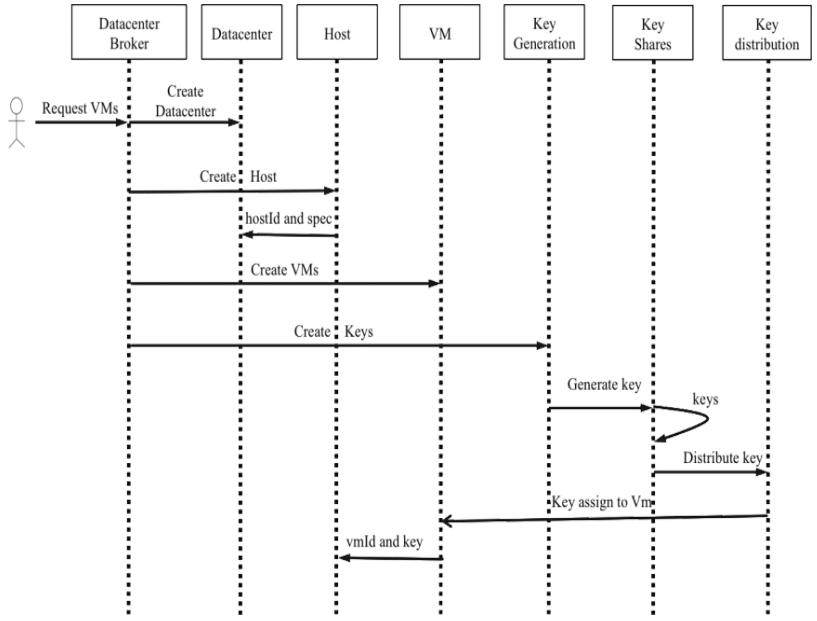

Figure 1. Key Generation and Distribution Process on the CloudSim

Figure 2 presents the process of key reconstruction on the CloudSim framework that consisting of Key Reconstruction class to collect the virtual machines (VMs) according to (*n, k*)-threshold scheme to create key back. The KeyReconstruction class is a new class that is added into the CloudSim framework. When the virtual mahine need to reconstruct the key, the Datacenter Broker class is responsible for collecting the *k* number of the virtual machines, which are relied on the same host and same data center. Moreover, the DatacenterBroker class is also extends to recombine the private key.

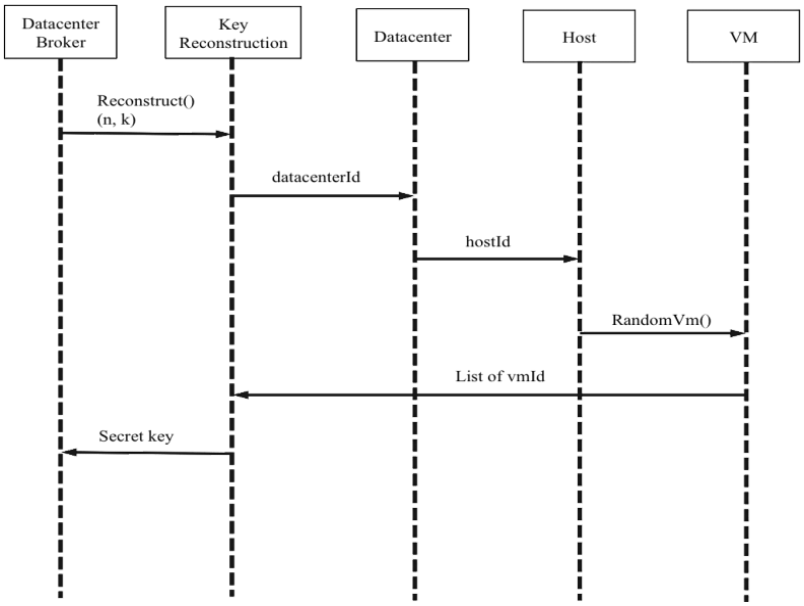

Figure 2. Key Reconstruction Process on the CloudSim

For environment setup, all the threshold cryptography processes were implemented using JAVA, including CloudSim, via NetBeanIDE 8.1 Application. All the experiments were run on Mac Os with CPU 2.4 GHz Intel Core 2 Duo, RAM 8 Gb. In cloud simulation, this experiment was simulated single host and sigle datacenter that consisting of environment parameters as shown in Table 1. The significant variable is the number of virtual machines, *n*, which equaled 32, 64, 96, 128, 160, 192, 224 and 256 respectively. In this experiment, moreover, the characteristics of each virtual machine are defined as same values.

Table 1. The Parameters of CloudSim to simulate Threshold Cryptography

Type	Parameter	Value
Data Center	Number of Data Center	1
	Number of Host	1
Host	Number of [1]PE	[3]{32, 64, 96, 128, 160, 192, 224, 256}
	[2]MIPS per PE	1000
	RAM (MB)	19200
	Size (MB)	1000000
	Bandwidth (MB)	10000
Vritual Machine	Number of VMs	{32, 64, 96, 128, 160, 192, 224, 256}
	Number of CPU per VM	1
	MIPS	250
	RAM (MB)	75
	Size (MB)	256
	Bandwidth (MB)	10
Cloudlet	Number of Cloudlets	{32, 64, 96, 128, 160, 192, 224, 256}
	Number of PE per Cloudlet	1
	MIPS	4000
	File size (MB)	300
	Output size (MB)	300

[1]PE = Processing Element, [2]MIPS = Million Instruction per second, [3]Variable to consider

As state earlier, this paper focuses on applying specifically the private key derived from RSA algorithm in threshold cryptography. Therefore, the private key will be split into the shares, distributed and reconstructed over the experiments. The key distribution and reconstruction time will be collected and evaluated in this paper to find suitable parameters. Here, that is threshold value, *k*.

3.1. Distribution Phase

The experiment in this phase is designed for observing, collecting and analyzing the time of key distribution by varying the threshold value, *k*, while the number of entities, *n*, is constant.

The key size used in the experiment will be only 2048-bit private keys according to the NIST recommendation [18].

The experiments were carried out by splitting the private key into *n* shares, where *n* was 256 shares, 224 shares, 192 shares, 160shares, 128 shares, 96 shares, 64 shares and 32 shares, respectively. The time taken to distribute the key shares was then recorded 10 times for same *n* shares and averaged next.

3.2. Reconstruction Phase

In reconstruction phase, the experiments were carried out as follows. First, after the private key was split into 256 shares, 224 shares, 192 shares until to 32 shares in order as in the section 3.1, the threshold value, *k*, was chosen by starting at the value of 8. It was then steadily incremented by 8 up to the maximum value of the number of shares, *n*. For example, if *n* were 256, the values of *k* used in the experiment would be 8, 16, 24, 32…until up to 256. The consuming time spent on reconstruction the private key based on the different threshold values, *k* was then recorded.

3.3. Finding a Suitable Threshold

In order to find a suitable threshold value, *k*, the graph of distribution time and reconstruction time will be plotted together. Accordingly, the intersection points between distribution time line and reconstruction time lines where the number of shares, *n*, was 256 shares, 224 shares until 32 shares and its each threshold value, *k*, varied along with 8 increment up to maximum value of its each share, n, in which the experiment design.

In this case, the average of each crossing point was claimed suitable threshold value, *k*, in terms of the time taken to both distribute and reconstruct. In other words, the number of threshold, *k*, is not too low

that it is easy for an adversary to obtain the key. At the same time, it is not too large, it will be taken so long to reconstruct.

4. RESULTS AND DISSCUSSION

This section presents the results of time collected in distribution phase and reconstruction phase of the 2048-bit private keys from section 3.1 and section 3.2. Then the suitable threshold value for cloud environment can be found according to the method proposed in Section 3.3.

4.1. Distribution Phase

The average times taken to share the sub-private key shares among *n* shares where *n* equaled 256 shares, 224 shares and 192 shares were presented in Figure 3. In order to analyse key distribution time for any shares, the forcasting analysis is computed to estimate such values by making the average.

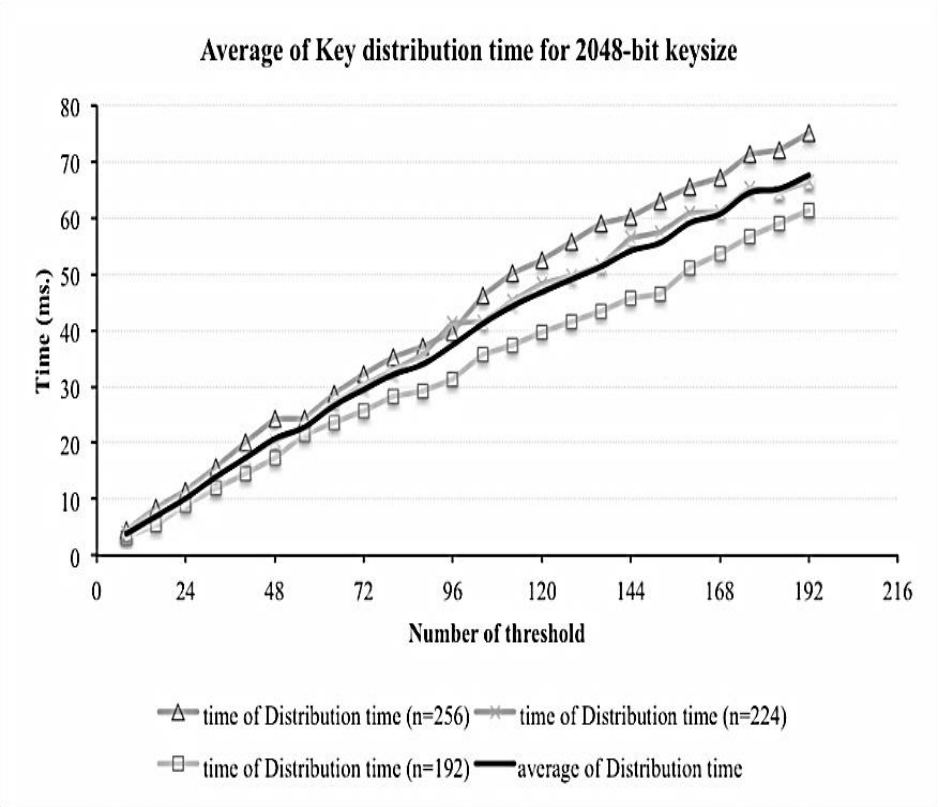

Figure 3. Avarage of Key Distribution Time for 2048-bit Keysize

Figure 3 illustrates that when the number of shares is 256, the distribution time is linearly increasing when the threshold value increases from 8, 16,…, 192. In the same manner, when the number of shares is 224 and 192, the distribution time progressively increases in an closely linear trends as the threshold value increasing.

On the whole, it can be observed that the average of the distribution time is stedily increasing when the threshold value increases, regardless of how many private keys are split into number of shares. This is used as a based tool to finding the suitable threshold value in the next section.

4.2. Reconstruction Phase

The average times of key reconstruction by using the threshold value, *k*, set in previous section were collected and simultaneously considerated with the average time of key distribution obtained from distribution phase as shown in the first column of Table 2.

From the results in experiment, eight groups of key reconstruction time divided by the number of shares is presented in Table 2. In the first column, the significant data is the average of key distribution time

which crossing with lines of key reconstruction time at the number of shares, *n*, equal 32, 64, 96, 128, 160, 192, 224 and 256 respectively. For highlighted data of each key reconstruction time is time interval harmonize with line graph of key distribution time crossing. This is the interval of number of threshold, *k*, when the number of shares, *n*, is 32, 64, 96, 128, 160, 192, 224 and 256 respectively.

In order to emphasize relation between key distribution time and key reconstruction time, set of line graph are demonstrated as Figure 4. From the graph, eight vertical lines are key distribution time devided by the number of shares, *n*, is 32, 64, 96, 128, 160, 192, 224 and 256 respectively. Each line is time to reconstruct the key that the threshold value, *k*, is steadily incremented by 8 up to the maximum value of the number of shares, *n*. Another is line of key distribution time which trends to be linear increment.

Table 2. Relation between Key Distribution and Reconstruction Time

Key Distribution time (ms.) (average)	Key reconstuction time (ms.)								
	nk	256	224	192	160	128	96	64	32
4.4	8	4.2	4.6	4.2	4.6	4	4	4.4	4.2
8.4	16	12.4	12.8	12.8	12.4	13.2	14.6	11.6	10.8
11.4	24	25.8	26	25.8	25.8	23.6	24.2	22	21.4
15.6	32	44.4	39.4	48.4	45	46	48	44	36.6
20	40	64.2	61.4	62.6	63.2	68.8	65	57	
24.4	48	82.8	79.2	85.2	83.4	90.8	88	83.8	
24.4	56	102.6	106.2	106.6	115.8	113.8	112	99.6	
28.8	64	125.2	130.6	133.8	142.4	147.8	150.8	126	
32.2	72	166.8	166.6	167	179.4	179	174.8		
35.2	80	206	196.6	203	199.6	202.8	200.8		
37	88	280.6	244	237.8	241.8	253.6	241.4		
39.6	96	345.6	285.4	280.4	290.2	291.4	302.2		
46.2	104	377.2	320.6	328.6	328.6	351.2			
50.2	112	436	364.4	370.4	399.4	400			
52.6	120	501.8	461	436.4	448.4	422.2			
55.8	128	538	497	500	497.4	488.2			
59	136	611	537.4	592.8	540.2				
60.2	144	654.4	615.2	638.8	608				
63	152	737.8	676.6	678.6	670.6				
65.6	160	831.8	760.8	787.4	783.4				
67.2	168	280.6	820.2	852.4					
71.4	176	897.8	910	915.4					
72	184	963	978.4	1003.6					
75	192	1034	1024.4	1065.8					
81	200	1121.2	1126.6						
81.4	208	1215.4	1211.2						
89	216	1321.8	1306.2						
92.4	224	1423.2	1395.6						
95.6	232	1541.8							
102.6	240	1664.6							
103.2	248	1760.4							
103.8	256	1872.6							

Using the results and graphs, the suitable threshold value can be found by looking for the intersection point between lines of key distribution time and key reconstruction time. This is the point where we believe to be such value.

The line graph of key reconstruction time at 256 keyshares, the cross point where k is around 56. That is approximately 22 per cent. The point of intersection, *k* value, where number of keyshares equally 224 and 192 are 52 and 44 by order so the percentage of threshold value for any keyshares is about 23. For the number of keyshares is 160, the intersection point of key reconstruction time and key distribution time where is threshold value point is 40 or just about 25 per cent. The cross point where is *k* value point is 36. That is, threshold value is approximately 28 per cent where the number of keyshares is 128. While the total of keyshares is 96, the point of intersection where *k* value equal 28 that is around 29 per cent. At the number of keyshares are 64, the crossing point at k equal 26 so the percentage of threshold value is about 41. The crossing point, threshold value is 17 where the number of keyshares is equally 32, the percentage of threshold value for any keyshares is around 54 per cent.

From the results, here, a method of finding an intersection point of key distribution time and key reconstruction time, a suitable threshold value, *k*, for all keyshares, *n*, is at approximately 31 per cent. It is, in this case, claimed as suitable value in term of time taken to distribute and reconstruct any keys. That means that using the lower number of thresholds, it is more easily for an adversary to obtain the key. In addition, it

is noticeable that the percentage of threshold value will be also increasing while the number of shares is decreased.

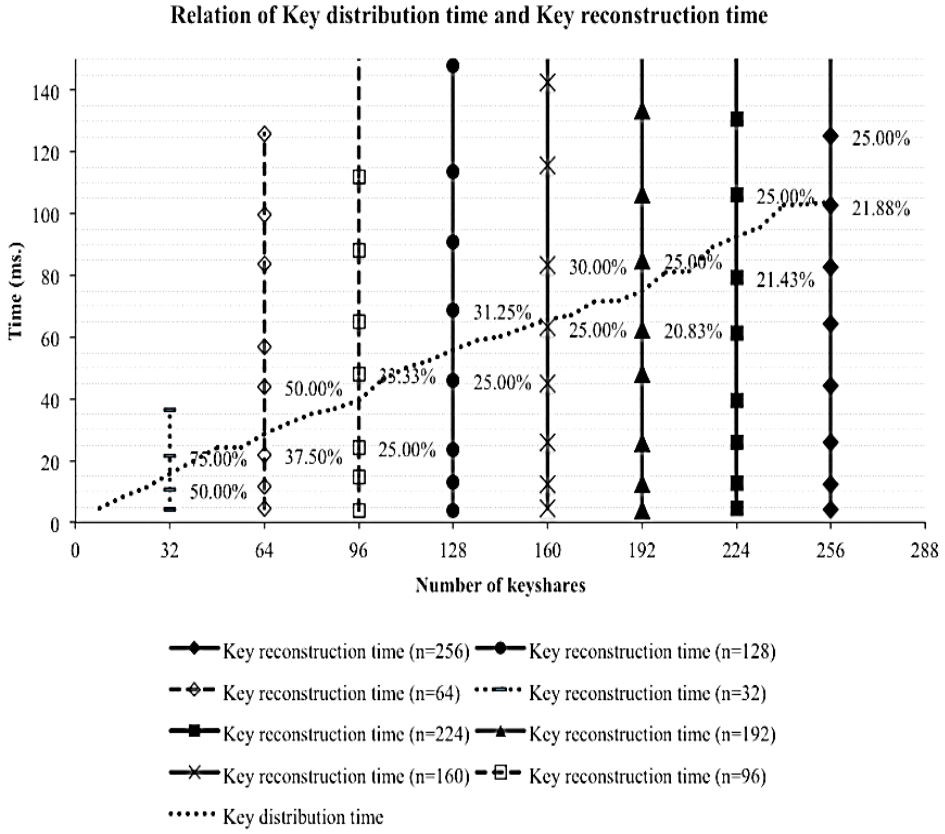

Figure 4. Relation of Key Distribution Time and Key Reconstruction Time (2048-bit keysize)

5. CONCLUSION

The protection sensitive data especially the private key in cloud computing is one of serious security issues. Therefore, threshold cryptography has been proposed to increase more security aspect. This scheme starts by spiting a private key into n shares then distributing them to each entity. When reconstructing key, only some of sub-key shares are collected. Anyway, the problem is that there are no any methods for finding a suitable threshold value, *k*, in terms of performances. Therefore, this paper presented a simple method by collecting time to share and reconstruct the keys. Using the collected data, they are plotted together as relative graph to find intersection point for achieving such value.

In the experiments, CloudSim is used as tool to simulate cloud environment for finding the time taken to distribute the shares to all entities and reconstruct the 2048-bit private keys by varying threshold value. Then, the suitable threshold value was able to find from the intersection point of two line graphs. It is claimed as suitable value in terms of time taken. The results from the experiments show that a suitable value should be 31 per cent of all shares, *n*. Although, this value is as the results from cloud simulation that is the limitation of the results, it can be used as a simple guideline for choosing a suitable threshold value, *k*.

REFERENCES

[1] NIST SP 800-145, "Recommendation for Key Management Part1: General (Revised)," *NIST Special Publication*, pp. 800-145, 2011.
[2] Right Scale, "State of the Cloud Report: Hybrid Cloud Adoption Ramps as Cloud Users and Cloud Providers Mature," *Right Scale*, 2016.
[3] Rivest R. L., *et al.*, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Communications of the ACM*, vol. 21, pp. 120-126, 1978.

[4] Kerckhoffs A., "La Cryptographie militaire," *Journal des sciences militaires*, vol. 9, pp. 5-83, 1883.
[5] Nikoval S., *et al.*, "Threshold Implementations Against Side-Channel Attacks and Glitches," *Proceedings of the 8th international conference on information and communications security, Springer-Verlag*, pp. 529-545, 2006.
[6] Demme J., *et al.*, "Side-channel Vulnerability Factor: A Metric for Measuring Information Leakage," *Proceedings of 39th Annual International Symposium on Computer Architecture (ISCA'12), IEEE*, vol. 40, pp. 106-117, 2012.
[7] Ristenpart T., *et al.*, "Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds," *Proceedings of the 16th ACM conference on Computer and communications security (CCS'09), ACM*, pp. 199-212, 2009.
[8] Dodis Y., *et al.*, "Key-Insulated Symmetric Key Cryptography and Mitigating Attacks against Cryptographic Cloud Software," *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS'12), ACM*, pp. 57-58, 2012.
[9] Mone S. P. and Dhotre S. S., "Enforcing multi-user security policies in cloud computing," *IJECE International Journal of Electrical and Computer Engineering*, vol/issue: 3(4), pp. 504-508, 2013.
[10] Alsolami F. and Boult T. E., "CloudStash: using secret-sharing scheme to secure data, not keys, in multi-clouds," *11th International Conference on Information Technology: New Generations (ITNG 2014)*, pp. 315-320, 2014.
[11] Cachin C., *et al.*, "Dependable storage in the intercloud," *Research report RZ 3783, IBM Research*, 2010.
[12] Xiong H., *et al.*, "CloudSeal: end-to-end content protection in cloud-based storage and delivery services," *Proceedings of SecureComm2011, LNICST. Springer-Heidelberg*, vol. 96, pp. 491–500, 2012.
[13] Pal D., *et al.*, "Multilevel Threshold Secret Sharing in Distributed Cloud," *Springer Communications in Computer and Information Science Series, CCIS: Security in Computing and Communications*, pp. 13–23, 2015.
[14] Bharill S., *et al.*, "A Secure Key for Cloud using Threshold Cryptography in Kerberos," *IJCA Trans. Cloud computing*, vol/issue: 79(7), 2013.
[15] Buyya R., *et al.*, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," *International Conference High Performance Computing Simulation*, pp. 1–11, 2009.
[16] Tani H. G. and Amrani C. E., "Cloud computing CPU allocation and scheduling algorithms using CloudSim simulator," *IJECE International Journal of Electrical and Computer Engineering*, vol/issue: 6(4), pp. 1866-1879, 2016.
[17] Pal S. O. and Pattnaik P. K., "A simulation-based approach to optimize the execution time and minimization of average waiting time using queuing model in cloud computing environmen," *IJECE International Journal of Electrical and Computer Engineering*, vol/issue: 6(2), pp. 743-750, 2016.
[18] NIST SP 800-57, "Recommendation for Key Management Part1: General (Revised)," *NIST Special Publication*, pp. 800-57, 2007.

BIOGRAPHIES OF AUTHORS

Weena Janratchakool is Ph.D. student at the Faculty of Information Technology, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. Her main area of interesting research is Cloud security. She received M.Sc. in Information Technology from KMUTNB.

Sirapat Boonkrong is an associate professor and an associate dean of academic and research affairs at the Faculty of Information Technology, King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. He received his B.Sc. and Ph.D. in Computer Science from the Department of Computer Science at the University of Bath, UK. His main area of research is information and network security. Previously, Sirapat worked as a researcher at National Electronics and Computer Technology Center (NECTEC) in Thailand. He also has experience in industry as a project manager at an IBM-partnered company. He is currently a full-time lecturer at the Faculty of Information Technology, KMUTNB and is also supervising several Ph.D. students all of whom are in the field of information and network security.

Sucha Smanchat is is currently a lecturer at the Faculty of Information Technology, King Mongkut's University of Technology North Bangkok, Thailand. He obtained his Ph.D. degree at Monash University in Melbourne, Australia. During his study, he was involved with the development of a prototype scheduler for Nimrod/K system. His current research interests are in cloud workflow scheduling techniques, MapReduce scheduling and data mining as cloud services.