International Journal of Security and Its Applications
Vol. 12, No. 5 (2018), pp.11-20
http://dx.doi.org/10.14257/ijsia.2018.12.5.02

Secured Data Partitioning through Sequence based Mapping and Random Order of Data Separation

Hazila Hasan[1], Suriayati Chuprat[2] and Mohd Naz'ri Mahrin[3]

[1]*Department of Information Technology and Communication, Politeknik Tuanku Syed Sirajuddin, Pauh Putra, 02600 Pauh, Perlis*
[2,3]*Advanced Informatics School (AIS), Universiti Teknologi Malaysia, 54100, Kuala Lumpur, Malaysia*
[1]*exult81@yahoo.com,* [2]*suriayati.kl@utm.my,* [3]*mdnazrim@utm.my*

Abstract

Data partitioning using secret sharing is a popular technique for securing data outsourcing in cloud computing. However, its complexity in reconstructing while preserving confidentiality has limited it for practical use. The drawback of secret sharing on how effectively reconstruct the secret shares, especially when it involves big data has motivated us to propose a sequence based mapping. Furthermore, in the current practice, the generated shares are being sent and stored in the original order which they are being generated. Thus, this could expose them to various threats, if attackers or curious server learn and observe the orders. Therefore, we have presented random order data separation to generate the random order of generating shares. This technique allows data to be separated into multiple chunks, and distributed to cloud storage in random orders. For evaluation, the proposed techniques have been evaluated through a series of simulation using maximum 10000 data. The performance was evaluated based on the time taken to achieve data reconstruction. As a result, we proved that sequence based mapping technique has improved the performance of data reconstruction compared to the indexing technique. In conclusion, sequence based mapping and random order of separated data are the ideal combinations for improving performance and preserving the confidentiality of data in cloud computing.

Keywords: *Data partitioning, Secret sharing, Cloud computing, Data security, Data privacy*

1. Introduction

Security has become the major concern in cloud computing. The reasons for this concern are due to rising of various attacks [1-3], threats [4-7] and issues concerning cloud computing availability [8-11]. Among the potential attacks that can occur or have occurred in cloud computing are wrapper attack in XML signature, denial of service attack and attack caused by multi tenant. The other potential attacks in cloud computing are packet sniffing attacks, man-in-the-middle attack, hypervisor attack, guest-hopping attack and SQL-injection attack. The mentioned attacks are not just perceived but they are real In addition, there are threats which give negative impacts to cloud computing and bad influence towards data security and privacy. The threats are data intrusion, malicious insiders and data loss or leakage Applying the encryption method as proposed by [12, 13] to handle the threats are useful because the data that are compromised cannot be retrieved and restored if the person does not have the knowledge of the encryption key. However, encryption does not protect the data from being deleted and also from malicious insiders. Malicious insider is the

Received (April 25, 2018), Review Result (July 25, 2018), Accepted (August 5, 2018)

ISSN: 1738-9976 IJSIA

internal threat which is the most dangerous and difficult threat, even though security solution such as passwords and encryption are implemented. This is because malicious insiders have more opportunity to obtain passwords and encrypted key without being noticed.

Adopting virtual private network, on the other hand, is beneficial to handle illegal communication interception, but still, it does not work when it involves malicious insiders. Due to the limitations in virtual private network and encryption method, many researchers have proposed data partitioning technique. Data partitioning is generally a technique of separating data (image, text, number) into smaller chunk [14]. Each chunk, then will be divided and distributed into different cloud storage [14] to secure the data. However, pure data partitioning alone is insufficient to handle the problem that caused by service unavailability due to the cloud's outage, cloud's server blockage or server failure. Thus, the urge for secured data partitioning have motivated researchers such as in [15-19] to propose data replication and secret sharing to be implemented with data partitioning.

2. State of the Art of Data Partitioning

The processes related to data partitioning divided into four. There are data separation, data distribution, data updating and data retrieval. Data separation is the process of dividing the data into two or more chunks. This process happens in the user's local machine. During this process, the criteria such as security and size of data are usually being considered. Data distribution will take place in allocating different data chunks into different clouds through a network. In this process the key factors, *e.g.*: Cost and cloud's quality of services will determine how data chunks will be allocated. The next process is data updating which will update the data that only occur if there are any changes in the data.

Data retrieval is the reverse process of data distribution and data separation. It's also known as data reconstruction. During data retrieval, security issue regarding cloud's blockage or outage has become the main concern.

Researchers have adopted the data partitioning for several reasons. For example, research done by [20] that is primarily to integrate mobile healthcare with multi cloud's environment has proposed data partitioning in order to increase the efficiency of data processing and to save energy cost [20] [21]. Based on similar motivation, Lee *et al.*, [21] are motivated to adopt data partitioning in the multi cloud to process big RDF Data. They have proposed a partitioning technique called SPA to increase the efficiency in partitioning big RDF graph data.

Research conducted by Zhao and Wang [15] is mainly due to the factor of maximizing cloud storage resources. In the research, data will be separated into chunks or fragment based on the user's preference. Once the fragments are created, metadata information known as fragment table that consists of fragment id, cloud id, username, password, partition attribute and vector interval is stored in the user's local computer. This table is created purposely to keep track of each fragment and its dedicated cloud storage. After that, the fragmented data will be allocated to different cloud storage based on the query generated. The problem with the query is that not all queries fit with this model and data partitioning might be skewed if the improper partitioning vector is used. Lastly, during the experimental analysis, the authors have mentioned that the increasing number of fragments will reduce the performance in terms of the space and time of cloud storage. In the paper, the authors have also proposed data replication to make sure data available.

In [16], the authors are focusing on a model to assist decision making during data distribution based on user's budget. The prime result of this model is the minimization of costs needed to be paid to the clouds' provider. This, however, has put aside other factors that may be critical and prioritized by some users or organization such as the performance of service providers, the amount of time data will be stored, security component provided, the ease of use of cloud services and many other user's requirements. In this model, the

size of data to be divided into the clouds is at the user's discretion. Therefore, without a proper method to be used, data division will become burdensome for some users. In addition, the absence of updating methods in this model has made the model be incomplete for the real environment.

Other researchers such as in [14, 17, 18] have focused on security reason for adopting data partitioning. They have conducted their research with the aim to ensure data confidentiality [14, 17, 18, 22] and data availability [17, 19]. The previously mentioned researches [15, 16] have also integrated security in their proposed works. The researches which are focused on security are named as secured data partitioning.

3. Secured Data Partitioning

Research done by Leistikow and Tavangarian [14] has explored data partitioning to secure picture sharing in the cloud. They proposed secured techniques called picture shredder which consist of image analysis, data separation and data distribution. The technique used facial recognition and stripping algorithm. Firstly, the image that is captured will be analyzed and sensitive information was identified. Secondly, images will be separated into two separate data (sensitive information and non-sensitive information). Finally, sensitive information will be distributed to Private Cloud and non-sensitive information will be distributed to the Public Cloud.

Zhao and Wang [15] have implemented the algorithms for all data partitioning process and proposed a model that allows each data chunk/fragment to be replicated and stored in the other cloud storage in order to ensure twenty-four hours (24) availability of data during the data retrieval process. This model was introduced to prevent data cloud's server blockage or server failure. However, according to Ye *et al.*, [19] data replication was a difficult approach, especially when it involves data updating. Furthermore, data replication has also required more storage space and increase processing time if big data involves. The model has also included data re-partition for data updating. Data updating will occur when the size of each chunk has grown. Nevertheless, the authors mentioned that "re-partition may lead to the division skew, which will lower efficiency of traversal query".

Research in [18] focused on ensuring data availability. This research proposed a model that takes advantage of partition-based cloud data storage. The researchers have come out with queries to communicate with clouds. It contains three processes of data partitioning which are data separation, data distribution and data retrieval. The data update method has yet to be explored further. Security technique solely implemented at the data retrieval process which is used to guarantee data availability. Data fragment has a master replication and the other clouds may store several different replications. This technique is the same technique applied in [15]. Thus, the same weaknesses as mentioned in [15] do exist in this model.

Research in [19] has been implemented to address the problems of data updating using lazy updates and consistency of verification in data partitioning. First, the storage servers were divided into server groups based on their location information. Within each group, the authors have applied short secret sharing to each data object and distribute the shares to servers in the group. Then the data shares are replicated to different groups. In this approach, the lazy update can be applied by updating shares only in one group and propagating the updates to other groups. Also, the consistent share verification can be performed within each group independently. Thus, according to the authors "the involved of server-to-server communications can be constrained within the group and the cost can be significantly reduced". However, the lazy updates technique which used lazy-group replication can perform well when it involves few clouds with simple transactions but may become unstable if the system scales up.

This model [22] has adopted vertical data partitioning algorithm to aid the separation process. The vertical partitioning algorithm works by dividing the data into different

fractions by splitting the columns or attributes of the database. However, a simple vertical data partitioning algorithm is not suitable for sensitive data. Therefore, the authors have adopted an encryption technique in their proposed model. Yet, the encryption technique does not protect against the malicious insider.

The research conducted by Singh *et al.*, [16] has implemented data separation and data distribution process through considering the cost and quality of service offered by each cloud service provider. For data retrieval, they have considered the problems of denial of service due to cloud service provider failure or security threats. In data partitioning, if one or more cloud storage failed to provide its service, the stored data cannot be reconstructed. Hence, denial of service may occur. Therefore, the authors have adopted a Shamir's secret sharing technique [23] in their model to ensure data availability during data retrieval. According to the authors, the adoption of the technique has allowed "at least q number of cloud service providers out of p number of service providers must take part to ensure a successful data retrieval".

In [17], the authors discussed the concept of encoding techniques to enhance data security of data before they are being distributed over different cloud storage. Firstly, the data was divided into small chunks by adopting information dispersal algorithm. By using the algorithm, the chunks will then be encoded into encoded symbols before they are stored in the multiple cloud storage. Once the data have been encoded, they will be stored in different clouds. To ensure the secrecy of data distribution, the authors have applied erasure coding techniques. To reconstruct the data, data that have been stored will be retrieved from the clouds. The retrieved data will then be decoded to get their original symbol. Lastly, the data will be combined to be the original file. In the data retrieval process, the same technique as implemented in [16] [23] were being used. The technique is prevalent since it does not require data to be retrieved from all cloud's storage, parts of the storage are sufficient to be used to reconstruct the data onto its original form.

Based on the related works, we could see that secret sharing is being adopted because of its flexibility during data retrieval. Only parts of the chunks are required in order to reconstruct the original data and accessing some data by illegal users will not expose the original data.

4. Proposed Works

4.1. Sequence based Mapping

To improve the performance of data reconstruction in big data, we propose sequence based mapping. The data from account_number field from the database will be read and later separated into chunks. For each chunk, a unique value/ index value will be assigned and a sequence value will be automatically generated. The sequence value of each chunk and its respective unique value will be mapped together to create a table. The mapping table will be stored in the user's local computer and will be accessed during data reconstruction. Figure 1 shows the steps involved in creating sequence based mapping. The difference between indexing technique [23-25] and our proposed work is that our work creates sequence value automatically together with the unique value meanwhile indexing only just create unique value called indexing as the reference for data reconstruction. The pseudo code for our proposed work is presented in Figure 2.

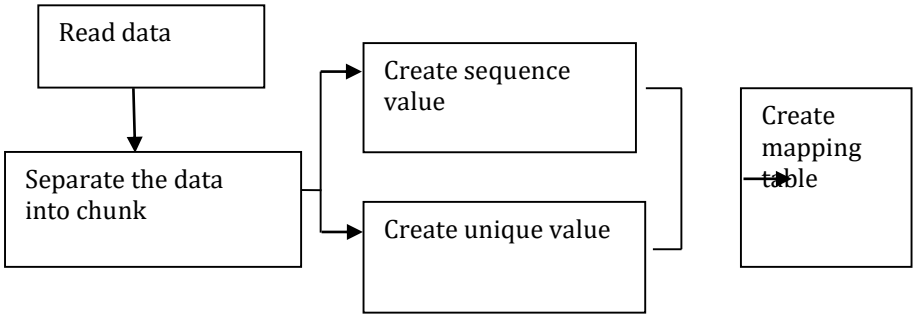

Figure 1. Sequence based Mapping Process

Algorithm for sequence based mapping

Read max_row //Maximum row required from database
For i = 0 to max_row

1. Read number_of_shares, number_shares_for_reconstruction
2. *Read account_number(i)*
3. *s = account_number, n = number_of_shares, k =number_of_shares_for_reconstruction*
4. *perform d = Shamirsecretsharing (s,k,n)*
5. *sequence_value = i // value will increase for each chunk*
6. *Read unique_value*
7. *value = i * unique_value*
8. *Write sequence_value, value // store in local computer*
9. *Write d, value // store in cloud's storage*

Figure 2. Pseudo Code for Sequence based Mapping

4.2. Random Order of Data Separation

To improve the security of data partitioning using secret sharing, we propose and add the random order of separated data. The data named account_number that have been read from the database will be separated into chunks. Each chunk together with its unique value will be reorganized into random order before they are being distributed to different cloud's storage. This technique is proposed to ensure that the related chunks (chunks that will be used to reconstruct to get original data) are not in the same order. Thus, this will improve the confidentiality of secret sharing. Figure 3 and Figure 4 illustrates the processes involved in creating the random order of separated data and Figure 5 presents the pseudo-code of the proposed work.

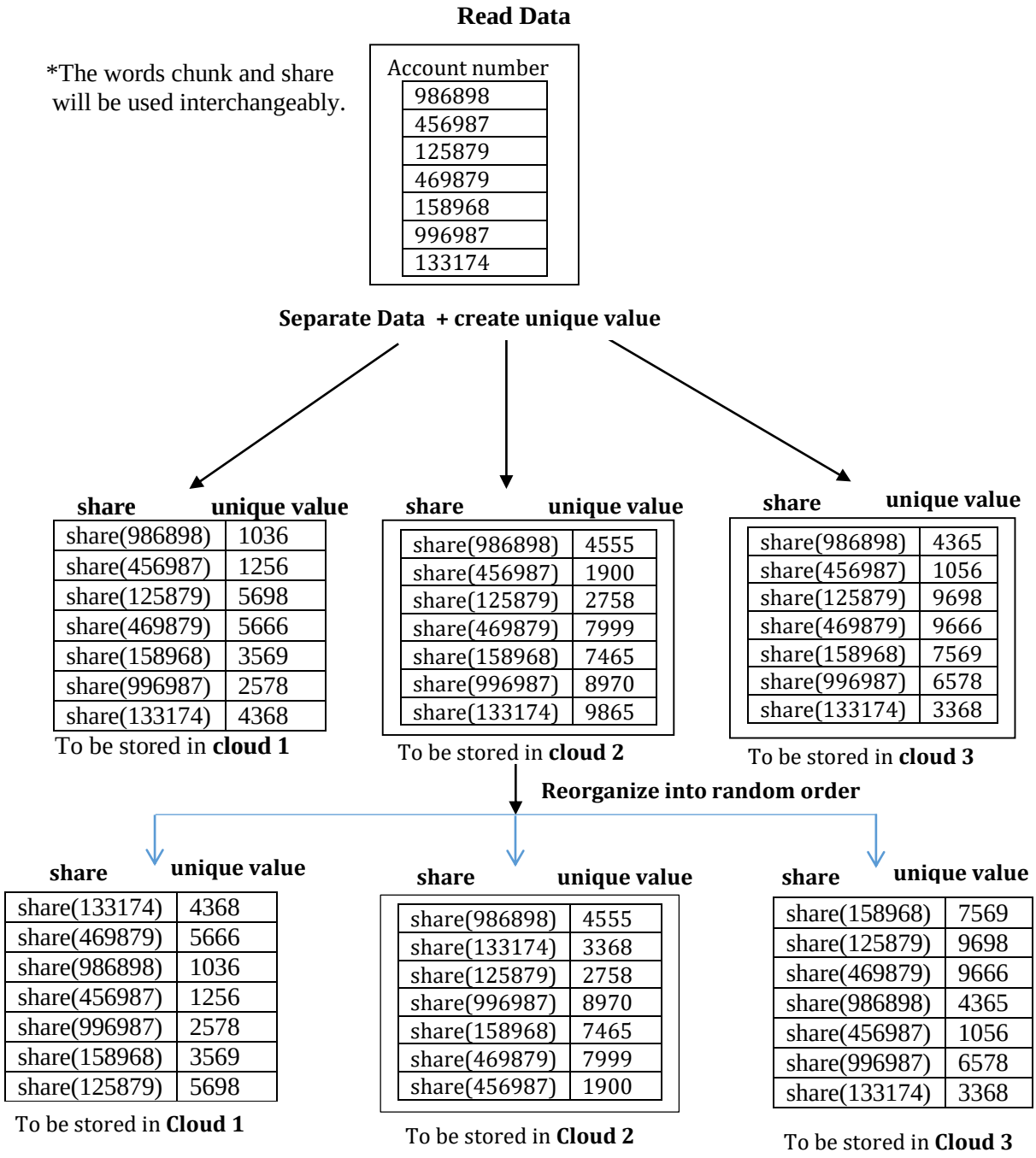

Figure 3. Illustration of Random Order of Separated Data

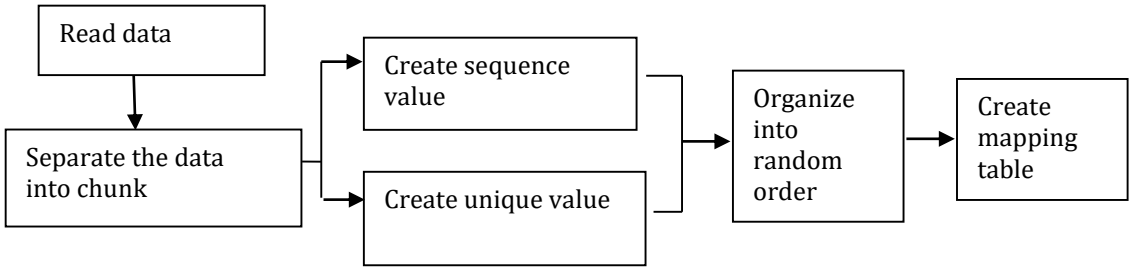

Figure 4. Processes Involved in Creating the Random Order of Separated Data

Algorithm for random order of separated data

1. Read d, value
2. *Write d, value into dataset Cloud*
3. *Read column_size_of_cloud*
4. *sizec1= column_size_of_cloud*
5. *Cloud_storage = Cloud*(randperm(*sizec1*),:) //*randperm is a procedure to randomly*

Reorganize the chuncks

Figure 5. Pseudo Code for Random Order of Separated Data

5. Methodology

The research is conducted to achieve the objectives on improving the security and speed of secret shares reconstruction especially when it involves big data. Thus, it has motivated us to propose a sequence based mapping with random order. The experimental evaluations are conducted using MatlabR2015b to write the algorithms. 10000 sample data (consists of 1 column and 10000 rows) are stored in MySQL database and used for experimental evaluation. The database is being connected to MatlabR2015b using MySQL connector ODBC. During the evaluation, the algorithms are executed and evaluated to compare the performance in terms of the time taken for data reconstruction for a series of data (starting from 1000 up to 10000) between the existing technique with the proposed technique (with and without the random order of separated data).

6. Results and Discussion

The following section will discuss on the results and discussion. The results of the reconstruction of the data chunks are from 1000 data into 10000 data are being collected and observed. Table 1 and Figure 6 show the result of indexing versus sequence based mapping without applying the random order of separated data in data chunk's reconstructing. The evaluation and comparison is made based on the performance, which is measured by the time taken to reconstruct the chunks of the different total number of data. It starts with 1000 account number and up to maximum 10000 account number. The results show that the proposed technique has improved the performance by reducing the time taken to complete the reconstruction of the chunks. From the observation, starting from 1000 data until 10000 data, the proposed technique has significantly reduced the time taken to complete the reconstruction of the data chunks.

Table 1. Result of Indexing Versus Sequence based Mapping (without random order of separated Data) in Reconstructing of Data Chunks

Total data	Time taken (in seconds) to reconstruct the chunks	
	Indexing	Sequence based mapping
0	0	0
1000	2.8036	0.43593
2000	6.2729	1.4513
3000	10.015	3.0332
4000	14.412	5.2716
5000	19.363	7.9898
6000	24.908	11.344
7000	31.075	15.221
8000	37.981	19.743
9000	45.178	24.84
10000	52.684	30.572

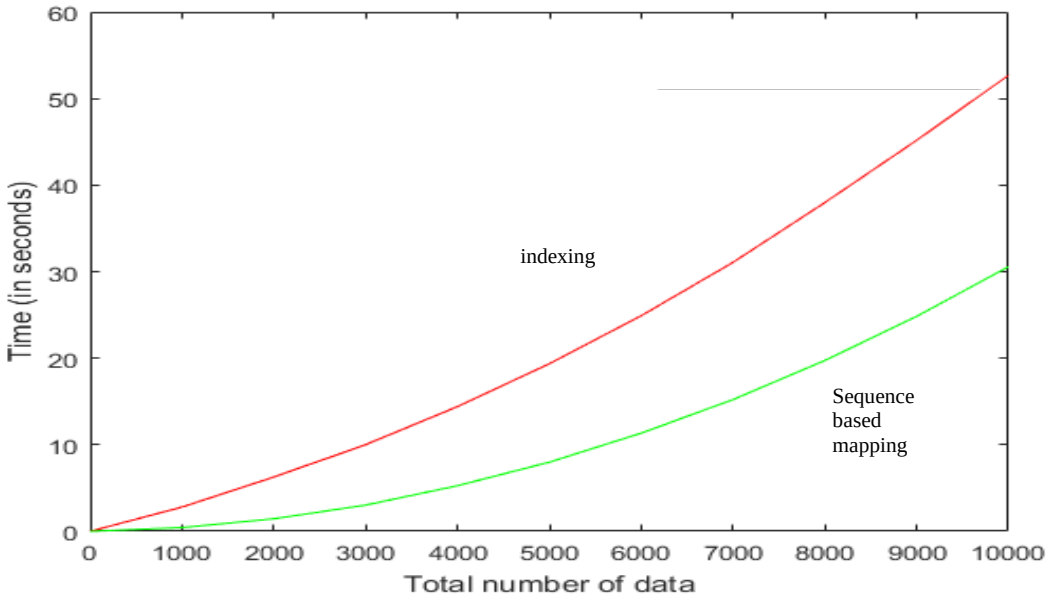

Figure 6. Graph of Indexing versus Sequence based Mapping (without random order of separated data) in Reconstructing of Data Chunks

The next result as shown in Table 2 and Figure 7 show the result of indexing versus sequence based mapping in data chunk's reconstructing after applying random order of separated data. Same as previous evaluation, this evaluation and comparison is also made based on the performance which is measured by the time taken to reconstruct the chunks for the different total number of data. It also starts with 1000 account number and up to maximum 10000 account number. After applying the random order of separated data, the data chunks are being reconstructed. The results show that the proposed technique has improved the performance by reducing the time taken to complete the reconstruction of the chunks. The proposed technique has significantly reduced the time taken to complete the reconstruction of the data chunks even though the chunks are being reorganized in the random order.

Table 2. Result of Indexing versus Sequence based Mapping (with Random Order of Separated Data) in Reconstructing of Data Chunks

Total data	Time taken (in seconds) to reconstruct the chunks	
	Indexing	Sequence based mapping
0	0	0
1000	7.377	0.47667
2000	8.2227	1.5832
3000	11.066	3.3498
4000	16.015	5.737
5000	21.371	8.7675
6000	27.511	12.429
7000	33.976	16.756
8000	41.301	21.689
9000	49.118	27.261
10000	57.455	33.502

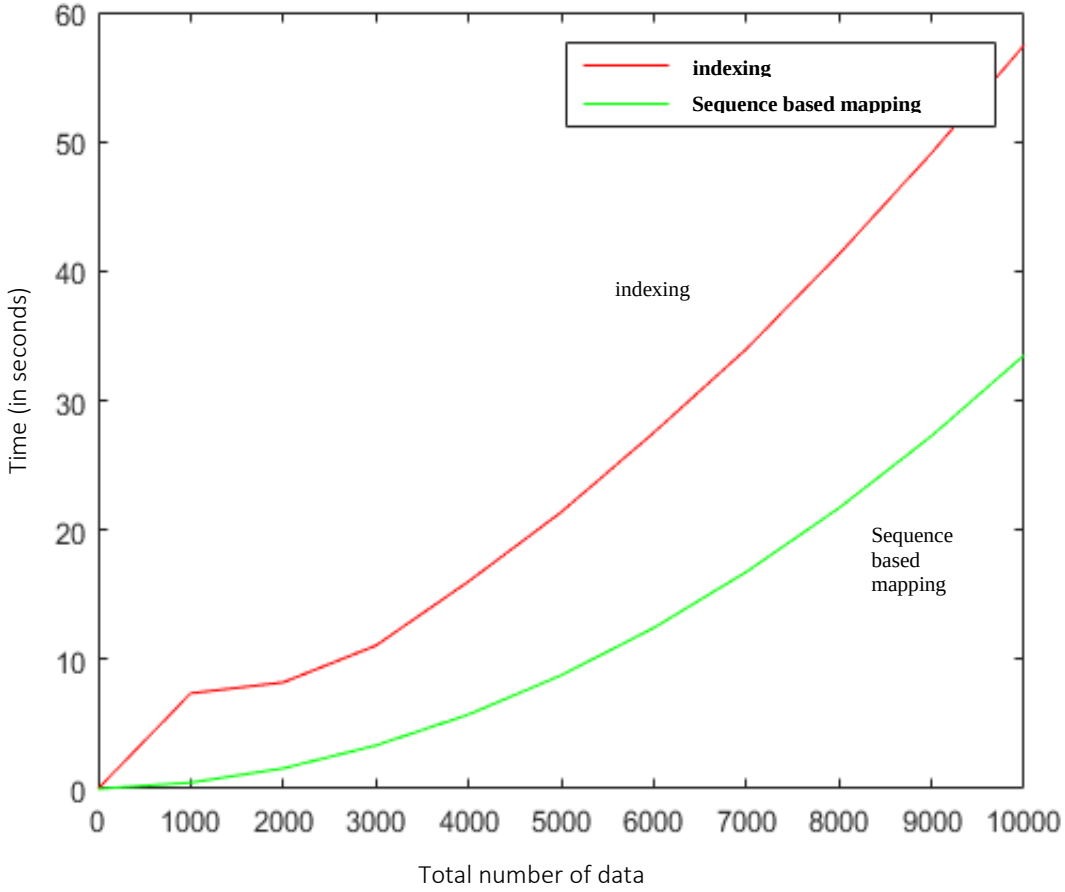

Figure 7. Graph for Indexing versus Sequence based Mapping (with random Order of Separated Data) in Reconstructing of Data Chunks

7. Conclusion

Our proposed work which is sequence based mapping has been proven to significantly improve the performance of time taken to complete data reconstruction in secret sharing. We are also presented random order of separated data to ensure the confidentiality of the chunks before being distributed to cloud storage. Therefore, we conclude that the sequence based mapping and random order of separated data are the ideal combination for improving performance and preserving the confidentiality of data in cloud computing.

Acknowledgements

This research was partly funded by Research University Grant of Universiti Teknologi Malaysia (Vote: 07462).

References

[1] F. Shaikh and S. Haider, "Security threats in cloud computing", Proceedings of the 2011 International conference for Internet Technology and Secured Transactions (ICITST), Abu Dhabi, **(2011)** December 11-14.

[2] C. John Roberts II and W. Al-Hamdani, "Who can you trust in the cloud?: A review of security issues within cloud computing", Proceedings of the 2011 Information Security Curriculum Development Conference, Kennesaw, Georgia, **(2011)** September 30 - October 01.

[3] K. Jamsa, "Cloud Computing SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security, and More", Jones & Bartlett Learning, Burlington, MA, **(2013)**.

[4] S. L. Garfinkel, "An evaluation of amazon's grid computing services: EC2, S3, and SQS", Technical Report TR-08-07, School of Engineering and Applied Sciences, Harvard University, USA, **(2007)**.
[5] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors", available on Techchrunch, **(2008)**.
[6] P. Boampong and L. Wahsheh, "Different facets of security in the cloud", Proceedings of the 15th Communications and Networking Simulations Symposium, Orlando, Florida, **(2012)** March 26-30.
[7] D. Hubbard and M. Sutton, "Top Threats to Cloud Computing V1.0", Cloud Security Alliance, **(2010)**.
[8] W. Kim, "Cloud Computing: Today and Tomorrow", Journal of object technology, vol. 8, no. 1, **(2009)**.
[9] M. A. AlZain, S. Ben and P. Eric, "Cloud Computing Security: From Single to Multi-clouds", 45th Hawaii International Conference on System Sciences, Maui, USA, **(2012)** January 04-07.
[10] D. Sinanc and S. Sagiroglu, "A review on cloud security", Proceedings of the 6th International Conference on Security of Information and Networks - SIN '13, Aksaray, Turkey, **(2013)** November 26-28.
[11] L. Wang, R. Rajiv, C. Jinjun and B. Boualem, "Cloud computing methodology, systems, and applications", CRC Press, Boca Raton, FL, **(2012)**.
[12] J. Yang and Z. Chen, "Cloud computing research and security issues", International Conference on Computational Intelligence and Software Engineering (CiSE), Wuhan, China, **(2010)** December 10-12.
[13] W. Liu, "Research on cloud computing security problem and strategy", 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Hubei, China, **(2012)** April 21-23.
[14] R. Leistikow and D. Tavangarian, "Secure Picture Data Partitioning for Cloud Computing Services", 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Barcelona, Spain, **(2013)** March 25-28.
[15] Y. Zhao and Y. Wang, "Partition-based cloud data storage and processing model", IEEE 2nd International Conference on Cloud Computing and Intelligent Systems (CCIS), Hangzhou, China, **(2012)** October 30-November 1.
[16] Y. Singh, F. Kandah and W. Zhang, "A secured cost-effective multi-cloud storage in cloud computing", IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, China, **(2011)** April 10-15.
[17] P. F. Oliveira, L. Lima, T. T. V. Vinhoza, J. Barros and M. Medard, "Coding for Trusted Storage in Untrusted Networks", IEEE Transactions on Information Forensics and Security, vol. 7, no. 6, **(2012)**.
[18] P. Ren, W. Liu and D. Sun, "Partition-based data cube storage and parallel queries for cloud computing", Ninth International Conference on Natural Computation (ICNC), Shenyang, China, **(2013)** July 23-25.
[19] Y. Ye, L. Xiao, I.-Ling Yen and F. Bastani, "Cloud Storage Design Based on Hybrid of Replication and Data Partitioning", 16th International Conference on Parallel and Distributed Systems, Shanghai, China, **(2010)** December 8-10.
[20] H. Wu, Q. Wang and K. Wolter, "Mobile Healthcare Systems with Multi-cloud Offloading", IEEE 14th International Conference on Mobile Data Management (MDM), Milan, Italy, **(2013)** Jun 3-6.
[21] K. Lee, L. Liu, Y. Tang, Q. Zhang and Y. Zhou, "Efficient and Customizable Data Partitioning Framework for Distributed Big RDF Data Processing in the Cloud", IEEE Sixth International Conference on Cloud Computing (CLOUD), Santa Clara Marriott, CA, USA, **(2013)** June 27-July 2.
[22] S. Subbiah, S. Selva Muthukumaran and T. Ramkumar, "An approach on enhancing secure cloud storage using vertical partitioning algorithm", Middle-east Journal of Scientific Research, vol. 23, no. 2, **(2015)**.
[23] S. Adi, "How to share a secret", Communications of the ACM, vol. 22, no. 11, **(1979)**.
[24] M. A. Hadavi, R. Jalili, E. Damiani and S. Cimato, "Security and searchability in secret sharing-based data outsourcing", International Journal of Information Security, vol. 14, no. 6, **(2015)**.
[25] I. M. Yusuf, Mustafa Kaiiali, A. Habbal, A. S. Wazan and S. I. Auwal, "A secure data outsourcing scheme based on Asmuth-Bloom secret sharing", Enterprise Information Systems, **(2016)**, pp. 1-23.

