Big Data Security on Cloud Servers Using Data Fragmentation Technique and NoSQL database

Nelson Santos, Giovanni L. Masala

Big Data Group, School of Computing, Electronics and Mathematics
Plymouth University, Plymouth PL4 8AA, United Kingdom
nelson.santos@students.plymouth.ac.uk
giovanni.masala@plymouth.ac.uk

Abstract. Cloud computing has become so popular that most sensitive data are hosted on the cloud. This fast-growing paradigm has brought along many problems, including the security and integrity of the data, where users rely entirely on the providers to secure their data. This paper investigates the use of the pattern fragmentation to split data into chunks before storing it in the cloud, by comparing the performance on two different cloud providers. In addition, it proposes a novel approach combining a pattern fragmentation technique with a NoSQL database, to organize and manage the chunks. Our research has indicated that there is a trade-off on the performance when using a database. Any slight difference on a big data environment is always important, however, this cost is compensated by having the data organized and managed. The use of random pattern fragmentation has great potential, as it adds a layer of protection on the data without using as much resources, contrary to using encryption.

Keywords: Cloud Security, Data Fragmentation, NoSQL Database, Big Data

1 Introduction

Cloud computing can be considered one of the most promising technology for IT applications. It is defined by NIST [1] as the model that enables on-demand access to a pool of resources (e.g., networks, storage, applications, and services) that can be rapidly provisioned with minimal effort from the service provider. This technology is growing in such a way that most modern applications are delivered as hosted services. Such services are divided into Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). This scenario has two main cornerstones: virtualization and distributed computing. They provide many benefits including terms of flexibility, elasticity and resource management. Big data is a big adept of this technology, as customers take advantage of the features offered to utilize and pay the resources needed to accommodate the business model and extend such resources when required [2]. This allows the customers to reduce the cost of the storage and computing clusters, as well deviate from the maintenance of the infrastructure and shift all the focus to the development [3].

Despite its benefits, cloud computing also brings many challenges. Among them, is the protection of the data and the privacy of the user. In cloud computing, the user's information is handed to the cloud provider and they are responsible for the storage and safekeeping of the data, often without disclosing their procedures to the end-user [2-5]. Furthermore, storing all the data with a single provider, along with the large number of mining algorithms available, leaves users susceptible to mining attacks from attackers with unauthorized access to the cloud and escalated privileges [6].
This paper investigates the use of random pattern fragmentation [7-8] on different cloud providers, to add a layer of security on the data, by measuring the performance to fragment, send and retrieve the data. In addition, a novel approach of managing the fragmented information on a NoSQL (Not Only SQL) database is proposed, with its performance also measured and compared. It will start by investigating the state of the art (Section 2), followed by the methodology in section 3. Afterwards, in section 4 the results will be displayed and discussed and compared to similar approaches, to provide a better evaluation of the performance, as well as a better understanding of the benefits and disadvantages of data protection by means of random pattern fragmentation.

2 State of the art

Encryption schemes present a satisfactory solution to the data privacy problem, however, they are very complex and computationally expensive [9-10]. Therefore, research has been shifting towards other alternatives. Kapusta et al [11] attempted to avoid encryption by splitting information on two distinct groups and provide different protection, according to the sensitivity of the data. Dev et al [6], approached the problem by categorizing and fragmenting data into chunks and store them in different providers, to avoid mining from providers, as well as attackers. Bahramim et al [9] proposed a lightweight modality for mobile phones, where random pattern fragmentation, based on chaos system, is used to split a JPEG file and store in multi cloud systems. Bahramim et al [10] investigates the use of databases to store and manage chunks created with the same method and adding a layer of encryption to the database. Lentini et al. [12] measured the performance of different fragmentation techniques on Amazon Web Services and compared them with the AES cryptography.

However, to improve the organization and overall management of the data in the server, it is imperative to use a database. Rafique et al [13] proposed a mapping strategy that leverages columnar NoSQL databases to perform data encryption at various levels of granularity dynamically. Alsirhanni et al [14] proposed a technique that stores data in different providers, by splitting into a master cloud that contains indexes of the fragments, and various slave clouds that store the data encrypted in columnar databases. Masala et a. [15] proposed data fragmentation on the cloud environment using a NoSQL approach, based on MongoDB [16] to take advantage of the highly scalable distributed architecture, which is the main characteristic of NoSQL.
The aim of this paper is the comparison of a novel approach (RPFNoSQLDB), having a mixed solution between a random pattern fragmentation approach and a NoSQL

database, with a random pattern fragmentation approach (RPF). The NoSQL solution adds a management layer on the scrambled data, offering therefore better scalability.

3 Methodology

3.1 Random pattern fragmentation

In the random pattern fragmentation (RPF), originally proposed by [9-10], but referencing the version implemented in [12], the original file is divided into N chunks and the pattern indexes are created with a random function, in other words, a random permutation of N elements before being stored in split files. The split files, are then saved on a cloud instance. The pattern indexes get stored in the client's machine, to reconstruct the original file when needed. With this technique, the attacker does not possess the knowledge of the random order and therefore cannot reconstruct the file. In the figure 1 the method is shown.

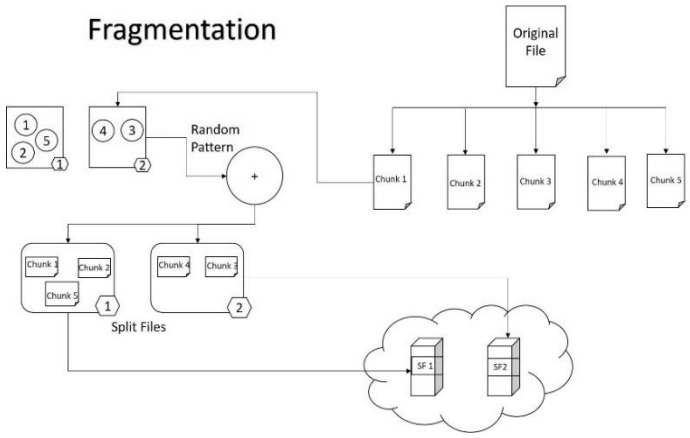

Fig. 1. The process of random pattern fragmentation. The original file gets split into chunks that are stored in split files. The split files are then saved on the cloud server.

In the reconstruction phase (Figure 2), the split files get downloaded from the cloud and reconstructed using the dictionary format, by combining the stored indexes on the client machine to the different chunks inside the split file. The chunks are then reshuffled back into the original order before being stored back into the client's device.

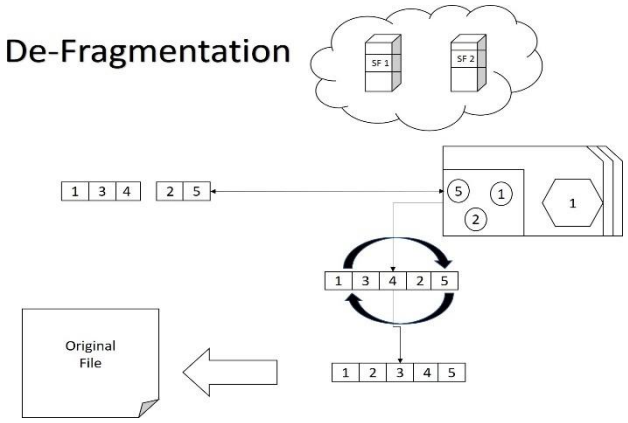

Fig. 2. The process of reconstructing the chunks back to the original file. The file is downloaded from the server and reconstructing using the indexes on the client's machine via a dictionary data structure. After the reconstruction, the file is stored on the client's machine.

3.2 The use of a NoSQL database combined with the random pattern fragmentation

We propose a novel approach (Figure 3) where we combine the use of the combination of the random pattern fragmentation with a NoSQL database (RPFNoSQLDB), where the original file gets split into chunks and those chunks are then inserted to split files.

The split files are then stored inside the NoSQL database that resides inside an instance on a cloud provider. The data is secured in transit with the use of the virtual private network (VPN) [19], and in case an attacker accesses the database, the chunks are in a random order, discouraging therefore any attempts to reconstruct the data. The details of the patterns are stored in the client's machine, which are then used to reconstruct the original file.

Using NoSQL to presents an advantage over relational databases, as the files are not structured, making the process of analyzing and retrieving the files faster. In the reconstruction phase, a method based on a dictionary is used, where the client machine uses the stored indexes, combined with the downloaded split files, to re-shuffle the chunks into the correct order, as shown in the figure 4.

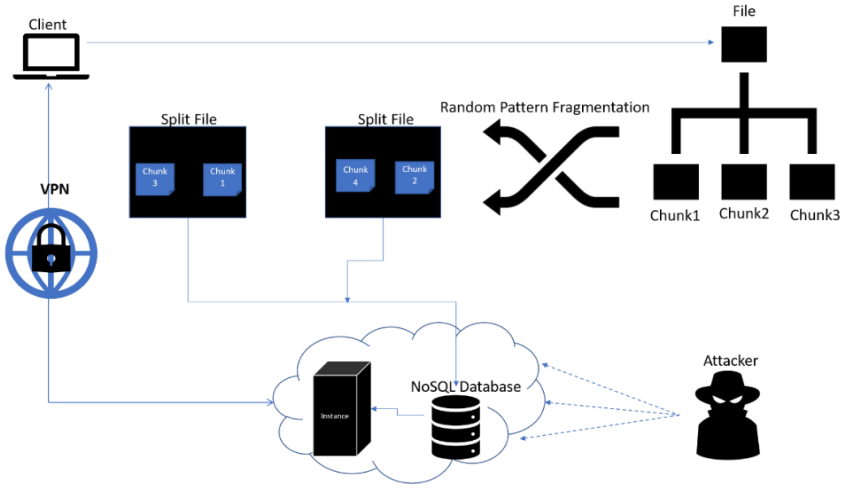

Fig. 3. Proposed model that uses random pattern fragmentation and stores the random chunks in split files, which are then stored on a NoSQL database.

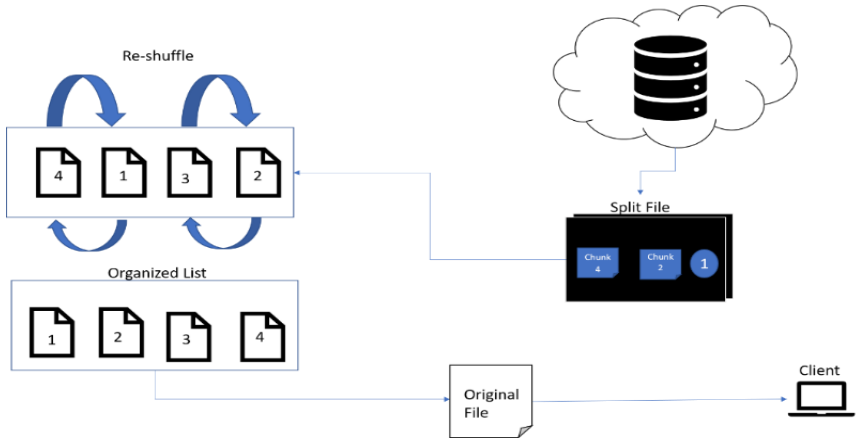

Fig. 4. Process of retrieving and reconstructing the original file. The chunks are sent from the database to the client via a VPN. In the client program, the chunks get re-arranged in a dictionary manner, where the client machine holds the indexes of all the chunks in the correct order.

4 Results and Discussion

In the first part of this paper we are aiming to analyze the performance of using data fragmentation on different cloud providers, as well as the performance of the connection type. This work investigates the performance of the most promising pattern fragmentation technique [12] in a virtual machine hosted by Amazon Web Services (AWS) [17], in comparison with the cloud offered by Microsoft Azure [18]. During the

investigation, we always consider sending the files to a single provider via a secure connection. The single provider is the worst-case scenario, as the entire data is available, providing a single point of attack for attackers to mine the data. Nevertheless, we are considering the typical scenario, related to the public cloud.

We are presenting different experiments, using the same algorithm and database in [12], with three different file types (.docx, .jpg, and .pdf), all with 100 KB of size. The result presented in [12] determines that the random pattern fragmentation is faster than the traditional AES encryption [20]. As a result, we are exploring the use of the random pattern fragmentation in the cloud environment.

In the first experiment, we test the random pattern fragmentation approach on a virtual machine in AWS [17] and Azure [18]. The time of splitting a file, storing in a virtual machine, retrieving and reconstructing back to the original file is compared between both providers, in figure 5. The communication between the client and the instance is done via tunnel-SSH. In addition, the time of sending a single .docx file, without fragmentation, is highlighted to compare the performance of using the fragmentation. It is visible, in figure 5, that Azure performs better than AWS, with an average of just above 1.5 seconds (i.e. considering also the sending of the original file without fragmentation).

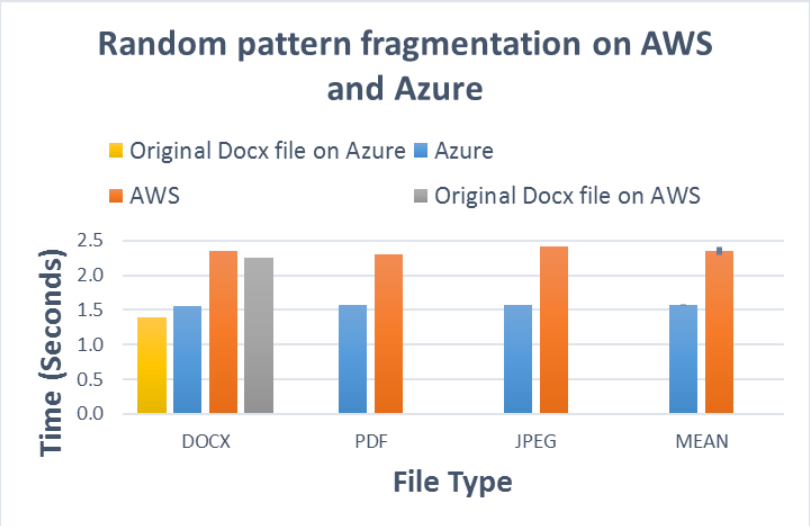

Fig. 5. Performance using tunnel-SSH on two different cloud providers. In the docx file is shown also the difference between sending the original file (called original DOCX) without fragmentation in both providers. On the mean bar is indicated also the standard deviation.

In the second experiment, we tested the proposed approach RPFNoSQLDB on two different scenarios, regarding the connection between the cloud and the client application. The chosen cloud environment to test the use of the database was Azure.

On one hand the program connected to the database using tunnel-SSH, and on the other hand the program interacted with the database using an encrypted Point-to-Site VPN. The results are displayed in Figure 6. tunnel-SSH displays slightly better results than its counterpart, however, given the standard deviation calculated in the mean, the difference can be considered neglectable. Nevertheless, using a VPN allows a clear communication channel between the cloud and the client, whereas with the ssh tunnel the client is opening a single connection to the host, complicating the process of transferring multiple files, as well as having multiple users on the application. In addition, with SSH the files are sent sequentially or with multiple connections from the same client, consuming therefore more resources from the server.

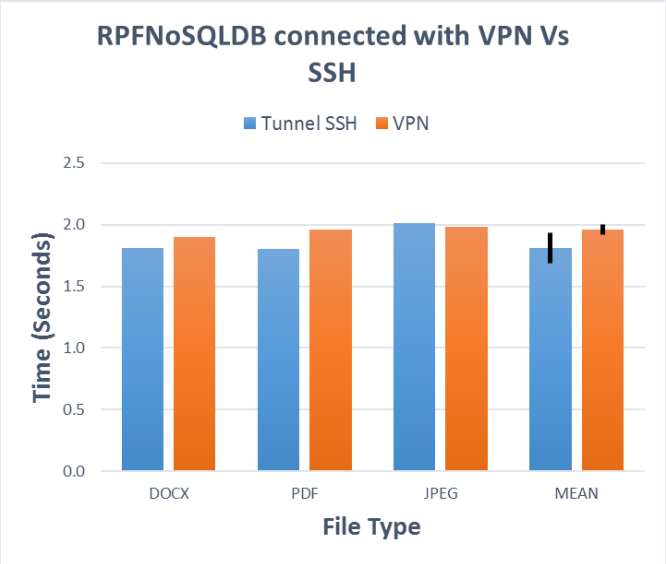

Fig. 6. Performance of the RPFNoSQLDB connecting Azure cloud with tunnel SSH vs VPN.

In the last experiment, using Azure cloud, in Figure 7, our proposed method RPFNoSQLDB was compared with the RPF, which does not contain a database. Further details are also published on table 1.

It can be derived from the figure 7 and table 1 that using a database to manage the fragments affects the performance. On the base of the first two experiments the results don't depend by the connection used (SSH or VPN). Such performance costs are relevant on a big data environment; however, this tradeoff compensates by having the data organized and structured, facilitating the management of the data.

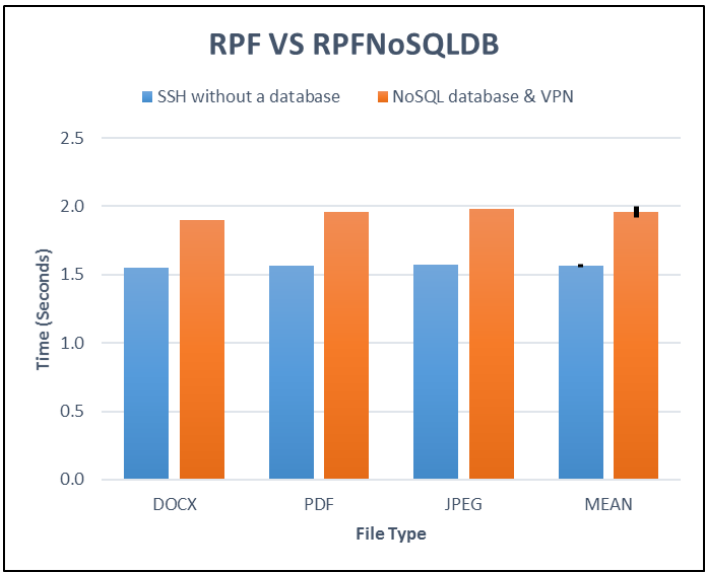

Fig. 7. Comparison of the proposed method RPFNoSQLDB (random pattern fragmentation + NoSQL database), which uses a VPN, with respect to the RPF (random pattern fragmentation without database), with uses a SSH connection.

Table 1. Evaluation of the performance of using RPFNoSQLDB with VPN over sending the files to the instance with respect to the RPF using a SSH connection. It encompasses the time to fragment the file, upload it, download and reconstructing the original file.

File Type	RPFNoSQLDB with VPN	RPF with SSH	Length Chunks
100 KB	**Time (Seconds)**	**Time (Seconds)**	**bytes**
DOCX	1.90	1.55	**1000**
PDF	1.95	1.57	**1000**
JPEG	1.98	1.57	**1000**
MEAN	1.96	1.57	**1000**
ST. DEV	*±0.04*	*±0.01*	**1000**

5 Conclusion

Cloud computing offers many advantages in terms of flexibility, scalability and reliability. Nevertheless, it also brings new challenges on security, data privacy and protection. We compared the use of splitting files and shuffling chunks on different cloud environments.

We also proposed a novel method of combining random pattern fragmentation and a NoSQL database (RPFNoSQLDB), to facilitate the organization and management of the data. When applying RPFNoSQLDB, trough the database structure, there is a trade-off on the performance, and the difference is compensated by having the data stored in an organized manner.

Furthermore, the use of a VPN creates a direct channel of communication between the client and the server, encrypted with IPsec, compared to SSH, where the different connections need to be created, to send the fragments without affecting the performance. Future work would include the use of columnar databases and storing the split files in different environments, and in binary large object formats, instead of using document-oriented databases, which store the information in JSON. These techniques show potential to the data security problem, as they add a further layer of security, without using many computing resources, which is not the case when traditional encryption methods like AES are applied.

6 References

1. NIST, Definition of Cloud Computing. National Institute of Standards and Technology (2011).
2. Cloud Security Alliance: Top threats to cloud computing. Version1.0. (2010)
3. Bahrami M. and Singhal, Mukesh. The Role of Cloud Computing Architecture in Big Data. Information Granularity, Big Data, and Computational Intelligence, Vol. 8.pp.275-295, Chapter 13, Pedrycz and S.-M. Chen (eds.), Springer (2015).
4. Kumar,P.Raj,H.Jelciana, P.Exploring Data Security Issues and Solutions in Cloud Computing,Procedia Computer Science 125,691-697 (2018).
5. Hegarty, R., Haggerty, J. Extrusion detection of illegal files in cloud-based systems. International Journal of Space-Based and Situated Computing 5(3), (2015)
6. Dev, H., Sen, T., Basak, M., Ali, M. An Approach to Protect the Privacy of Cloud Data from Data Mining Based Attacks. In: Companion: High Performance Computing, Networking Storage and Analysis 2012, pp.1006-1115 IEEE, Salt Lake City (2012)
7. Chakraborty, Debrup, and Palash Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. Fast Software Encryption. Springer Berlin Heidelberg (2006).
8. Gharajedaghi, Jamshid, Systems thinking: Managing chaos and complexity: A platform for designing business architecture. Elsevier (2011).
9. Bahramim M. and Singhal,M. A Light-Weight Permutation Based Method for Data Privacy in Mobile Cloud Computing. In 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, pp. 189-198. IEEE, San Francisco (2015).

10. Bahrami, M. and M. Singhal, M. CloudPDB: A light-weight data privacy schema for cloud-based databases.In: 2016 International Conference on Computing, Networking and Communications (ICNC). pp. 1-5. Kauai (2016).
11. Kapusta, K. and Memmi, G. Data protection by means of fragmentation in distributed storage systems. In: 2015 International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS). pp. 1-8. IEEE. Paris (2015).
12. Lentini S, Grosso E, Masala G, A comparison of data fragmentation techniques in cloud servers**.** In**:** proceedings of International Conference on Emerging Internet, Data and Web Technologies (EIDWT 2018), Tirana **(**2018**)**.
13. A. Rafique, A. Landuyt, D. Reniers, V and Joosen, W. Leveraging NoSQL for Scalable and Dynamic Data Encryption in Multi-tenant SaaS. In: 2017 IEEE Trustcom/BigDataSE/ICESS. pp. 885-892. Sydney (2017).
14. Alsirhani, A., Bodorik P. , and Sampalli, S. Improving Database Security in Cloud Computing by Fragmentation of Data, in International Conference on Computer and Applications, PP-43-49. IEEE.Dubai (2017).
15. Masala G.L., Ruiu P., Grosso E. Biometric Authentication and Data Security in Cloud Computing. In: Daimi K. (eds) Computer and Network Security Essentials. Springer (2018).
16. MongoDB Homepage, https://www.mongodb.com/, last accesses: 2018/03/03
17. AWS Amazon Homepage, https://amazon.com, last accessed 2018/03/02.
18. Microsoft Azure Homepage, https://azure.microsoft.com/en-gb/, last accessed 2018/03/02.
19. SANS Institute: Extending your business network through a virtual private network (VPN). SANS Infosec Reading room. (2016)
20. Federal Information. Announcing the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197 (2001).