Asian Journal of Information Technology 15 (20): 4102-4105, 2016
ISSN: 1682-3915
© Medwell Journals, 2016

Effective Storing and Managing of Data in a Multicast Communication Over Cloud Environment

Lino Abraham Varghese and S. Bose
Department of Computer Science and Engineering, College of Engineering Guindy,
Anna University, 600025 Chennai, India

Abstract: Cloud computing has been emerged as a pla tform for exchange of the data in a multicast communication and to store the data over a period. Particularly in the case of medical data which may be sensitive information, the privacy of data is a primary concern. If left unprotected these could bring forward grave consequences on privacy. The common solution is to encrypt the data before placing it in the cloud. Some prominent risks include key management, user revocation and freshness of data. Here the data were encrypted and stored in cloud to get the benefits of the cloud while key for encryption is stored at the client side. The data are stored in the cloud till its retention period is getting over. For the effective computation of encrypting the data and to check the retention period of data that were stored in the cloud a Hadoop frame work is used.

Key words: Multicast communication, data retention, Hadoop, forward secrecy, backward secrecy, cloud storage

INTRODUCTION

The computing paradigm has evolved substantially from the mainframes which are shared by different users. Then came the personal computer or desktop computer used by a single user. The individual systems are connected to form the local area networks. In the next era of development, the local area networks are interconnected which leads to the formation of Internet. Now the phase of computation changed to distributed computing from the single processor. Finally, now it reached a state to access scalable unlimited amount of storage and computing by opening the way to cloud computing.

Cloud computing usually provides different levels of services such as Infrastructure as Service (IaaS), Platform as Service (PaaS) and Software as Service (SaaS). In a multicast communication, the cloud vendors can provide infrastructure to store the data that are communicated across the network. Through multicast communication data can be effectively and efficiently communicated between groups of users. The IT infrastructure is provided to the end user by the cloud provider over the Internet. If a cloud is used, the end user can forget about the infra structure maintenance cost and concentrate on the business logic. Once the data are in cloud the end user or the data owners have only limited control over the data as well as the infra structure on which the data are stored. The end user expects from the cloud vendors that whatever data stored, it should be secure and it should be available at all time.

To secure the data in cloud, the stored data are encrypted and making it unavailable to unauthorized users. There are cases (Ateniese *et al*., 2007) where the Cloud Service Provider (CSP) may hide data loss incidents to maintain the reputation of the firm. Sometimes rarely accessed data (Juels and Kaliski, 2007) may be discarded by the CSP. To avoid unnecessary download and upload of data to check the correctness of data local authentication is avoided. The stored data are properly audited by a third party auditor. In the multicast communication key management should be properly done. The key has to be communicated between the users in a secured channel.

Here a dynamic closed group of multicast communication is considered. Since the group is dynamic at any moment of time the group members can be added or evicted from the group. Therefore the system should maintain forward and backward secrecy. Outsiders are not allowed to communicate to the group; hence the group is a closed one. There are mainly two types of users in a group, one is the group admin and the other is ordinary member. The group admin is responsible to add or terminate any member to the group. When a new user wants to join the group the user will send a request to the group admin and with the authentication protocol the user

Corresponding Author: Lino Abraham Varghese, Department of Computer Science and Engineering, College of Engineering Guindy, Anna University, Chennai 600025, India

is allowed or denied to the group. The key have to be changed whenever a change happens to the group. This is phenomenon is known as rekeying. For the faster computation process Hadoop frame work is used. The stored data over the period of time become irrelevant and have to be destroyed. The retention of the data in the cloud depends on the retention policies that are set. The retention policy mainly depends on individual files, organization and state or country policy. Prompt removal of data reduces the storage cost and helps in effective management of sensitive data.

Literature review: The data stored in the cloud have to be deleted permanently after the retention period is over. To make the data unrecoverable, the original data is overwriting (Gutmann, 1996) with new data repeatedly. In the cloud environment the data are stored somewhere in the vendor's premises. There is no guarantee that the original data is overwritten. This technique may not be suitable for data that are stored in the cloud.

Using time based control key, the key manager will remove the key used to encrypt the file over the period of time, which is specified during the creation of file. The key is destroyed and theoretically that data will become inaccessible. So in general even if the file is not removed the file will be in an encrypted stage. There is no quantitative evaluation given for this procedure. With a self destruction data system (Zheng *et al*., 2012) promotes privacy of data which will cause the destruction of data especially sensitive data without an initiation from the user part. But here the time consumption of data retrieval and decrypting the data is high and the replicated copies may not be deleted promptly. FADE (Tang *et al*., 2012) provide security over the outsourced data stored at third party premises. Each file is associated with file access policies. The outsourced file are encrypted using cryptographic key usage.

MATERIALS AND METHODS

In the multicast communication, by storing the data in cloud the end user can enjoy the benefits of the cloud such as confidentiality, authenticity, availability and freshness of data. Our work addresses the problem of confidentiality and retention of stored data on cloud environment.

The stored data should be highly confidential. Unauthorized persons are not allowed to participate in the group communication. Stored data have to be encrypted using symmetric keys. Regarding the data that are circulated the forward and backward secrecy should be maintained. Forward secrecy means an expelled group member or member who left the group are denied to access group communication. Backward secrecy means new member joined the group is prevent from decoding messages exchanged before he joined. The session keys are issued and distributed by the group admin. The key distribution protocol is intended to ensure both backward and forward secrecy. Encryption of data provides resource protection while key management enables access to protected resources. Figure 1 shows the system architecture. Algorithm 1 and 2 shows how the system will respond when a new user tries to enter the group or a user leaves the group, respectively.

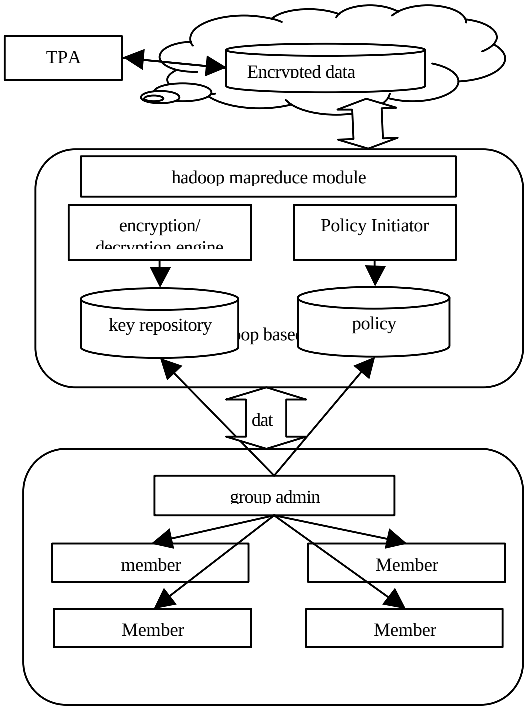

Fig. 1. System archtitecture

Algorithm 1: User Join

```
STEP 1: New user sends a REQ message
STEP 2: Group admin initiates Authentication Protocol
STEP 3: Begins Authentication Protocol
STEP 4: Check the credibility of the user
STEP 5: On satisfaction, allow the user to join the group
STEP 6: End
STEP 7: Group admin initiates Key Management Protocol
STEP 8: Begins Key Management Protocol
STEP 9: Send the new key across the group
STEP 10: End
```

Algorithm 2: User left

```
STEP 1: User sends a left message or Admin sends a revoke message
STEP 2: Admin revoke the user privileges
STEP 3: Group admin initiates Key Management Protocol
STEP 3: Begins Key Management Protocol
STEP 4: Send the new key across the group
STEP 5: End
```

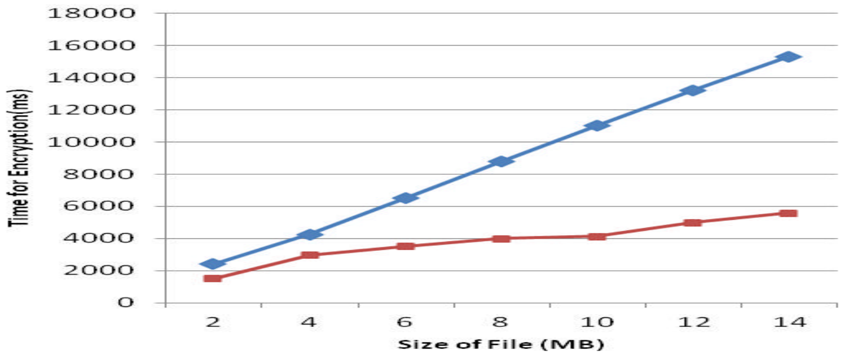

Fig. 2: Time for encryption V/s size of file

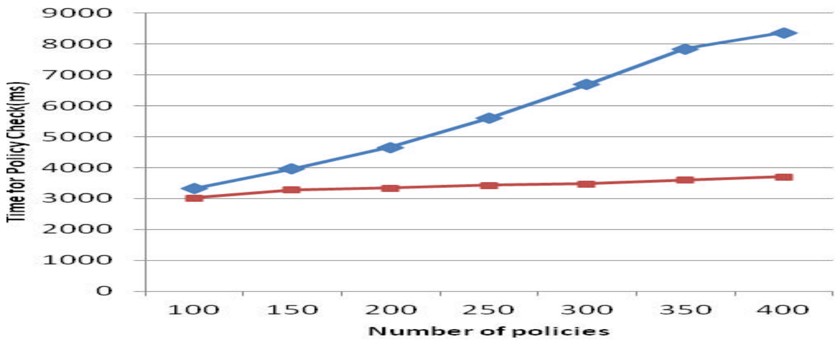

Fig. 3: Time for policy check V/s No. of policies

Hadoop frame work contains the distributed file system called the HDFS and scalable data processing part called the MapReduce (Dean and Ghemawat, 2008). The system was deployed on 4 machine linux cluster of 32 GB RAM and loaded with Java1.8.0_31 and hadoop1.2.1. Out of the four machine in the cluster one machine is configured as master node and others slaves. A one hundred retention policy were set and populated in the policy repository. The files of different MB size were given to the encryption/ decryption engine. Encrypted data are stored on amazon simple storage service (Amazon S3). Secure and scalable cloud storage can be achieved by storing the data over Amazon S3. There is no installation cost for S3 and runs on pay per use model. The data may be replicated and stored at different data stores to achieve the high availability of data based on the service level agreement. The policy initiator initiates to enforce the policy on the file and as well as validate the policy at regular interval of time. Once the expiry of time reaches, the file and all the meta data of the file as well as the keys used to encrypt the file are also deleted. The integrity of the encrypted data on the cloud are checked by Third Party Auditor (TPA) using homomorphic encryption. This architecture ensures the user high durability and high integrity of data at a low communication cost.

RESULTS AND DISCUSSION

Retention policy states about how long data have to be kept and which all data have to be stored. We have used the same retention policy for file based system and hadoop based system. Figure 2 shows the graph use of time for encryption plotted against the size of the file. Results were evaluated for different size of file and time for encryption was calculated. It was found that for small size file, hadoop based system may not much efficient, but as size increase the efficiency is tremendous. We have conducted experiments by changing the number of policies on fixed number of files which is having the size of 2MB. Figure 3 show graph usage of time for checking retention policy plotted against the number of policies. As the results shows as the number of policies grow high, the hadoop based system is having high upper hand over the file based system. Both the graph shows the efficiency of hadoop based sustem over the file management system.

CONCLUSION

A prototype of data retention and management of the users in a multicast communication was created. It is found that as the retention period is over the files stored on the cloud were deleted without user intervention. The retention policies and the key were kept in the client side and avoiding the chance of cloud provider to do anything wrong. The encrypted data is stored on the cloud, so that the user can access the data at any moment of time. With the rekeying mechanism, we have made sure that forward and backward secrecy is maintained. This prototype can be extended by adding the freshness concept of data on a high availability environment by keeping the data at multiple locations is worthy of further exploration.

REFERENCES

Ateniese, G., R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson and D. Song, 2007. Provable data possession at untrusted stores. Proceedings of the 14th ACM Conference on Computer and Communications Security, October 29, 2007, Alexandria, Virginia, USA., pp: 598-609.

Dean, J. and S. Ghemawat, 2008. Map Reduce: Simplified data processing on large clusters. Commun. ACM., 51: 107-113.

Gutmann, P., 1996. Secure deletion of data from magnetic and solid-state memory. Proceedings of the 6th USENIX Symposium on Security Symposium, July 22-25, 1996, USENIX, San Jose, California, pp: 77-90.

Juels, A. and B.S.Jr. Kaliski, 2007. PORs: Proofs of retrievability for large files. Proceedings of the 14th ACM Conference on Computer and Communications Security, October 29-November 2, 2007, ACM, New York, USA., ISBN:978-1-59593-703-2, pp: 584-597.

Tang, Y., P.P.C. Lee, J.C.S. Lui and R. Perlman, 2012. Secure overlay cloud storage with file assured deletion. IEEE. Trans. Dependable Secure Comput., 9: 903-916.

Zeng, L., S. Chen, Q. Wei and D. Feng, 2012. Sedas: A self-destructing data system based on active storage framework. Proceedings of the IEEE Conference on APMRC 2012 Digest, October 31- November 2, 2012, IEEE, New York, USA., ISBN: 978-1-4673-4734-1, pp: 1-8.