Extended and Modified Fade for Self-Destruction of Data in Cloud

K. Pradeep kumar, G. Lakshmeeswari

Abstract— Cloud Computing is a technology that offers many services through Internet and remote servers. Data storage is one amongst its services. Storing voluminous data generated as a part of day to day business activities requires loads of storage resources and is costly affair to own these storage devices. A better solution to this is Cloud Storage. Data stored in cloud is made available as a service through a network. Cloud offers data access irrespective of the user or data location with minimal efforts. This flexibility has attracted many users towards cloud storage services. Data gets transmitted through internet and users can access data regardless of their location or access device. It is the combined responsibility of the user as well as the Cloud Service Provider(CSP) to safeguard data on the cloud.

Service Level Agreement(SLA) is a contract between the provider and the user that contains the terms and duration of service offered by the cloud provider. During the service period, Multiple copies of data is backed up at different geographical locations for data availability and every trace of data may not be completely destroyed on expiry of SLA. Even after deletion of user's data from cloud there may be some data which is not visible to the user but may be present on servers. This is referred as Data Remanence. When a file is deleted the Operating System(OS) will remove the file entry in the file system but data will exist on the physical hard drive in the data area or data block and can be recovered using retrieving tools. A solution to this problem has been proposed in order to overcome this by overwriting the actual file contents.

Keywords: Cloud computing, Cloud Storage, Data Remanence, Service Level Agreement, FADE, Encrypt-Overwrite, Key authority.

1. INTRODUCTION

Cloud is the single stop solution for users to procure services on demand. The CSP will provide these services to the user on pay for use basis. The services are provided to the user as per SLA. SLA is a contract between the CSP and the user regarding the services, quality and tradeoffs provided by the CSP to the users. Once the SLA expires, the user's data is deleted from cloud.

There are number of ways which are followed by CSP to delete data of the user after the expiry of the SLA. Many of the cloud service providers use garbage collection to delete data. The deleted files are being marked and send to a separate disk and this data is available there until the expiry of retention period. Data retention depends upon data classification, where highly sensitive data may be retained for longer durations than less sensitive data. After the retention period, the data on that disk has is wiped permanently. In this case we need to completely depend upon the cloud service providers for deletion assurance. Self destruction of data was proposed by researchers to minimize the dependency on CPS's for data deletion assurance. Vanish, Safe vanish, FADE, SFADE, SFADE+ etc methods were proposed for this purpose.

2. LITERATURE SURVEY

Self-destruction of data is counter measure for the prolonged availability of data even after expiry of SLA. Self-destruction mechanism mainly focuses on protecting user data privacy. The important and sensitive data gets destroyed after a particular time period.

Vanish is a method for destructing the data in cloud. It uses Distributed Hash Tables(DHTs) which has the property of discarding data after certain time. Each key is encrypted with a random key and the key shares are stored in large public DHT. The limitations with vanish are hopping attack and sniffing attack. In Hopping attack the network is attacked by sending packets to port that is usually inaccessible from a given end system. Sniffer attack is that in which a sniffer can monitor, read and capture network data exchange and read network packets. If the packets are not encrypted, a sniffer furnished a complete view of the data available in the packet. Hopping attack can be overcome by moving to a privately hosted DHT and sniffer attack can be overcome by using RSA a encryption algorithm during transmission [1].

Safe vanish is another method for deletion of files on cloud. But Sybil attacks can efficiently recover 99% of the vanish messages. So as a counter measure the vanish keys are stored on both private DHT and a public DHT. The problem arises if both the DHTs are insecure, the length of the key shares are increased making the attacks very costly. The availability of key shares depends on DHT's which have the property of discarding data after certain period leading to loss of key shares. The limitation of safe vanish is that the length of the key shares must have a limit[2].

File Auto Deletion (FADE) means that files are made permanently in accessible. The encrypted data remains with the CSP and the encrypted keys with the trusted key manager. FADE specifies policy based, time based file assured deletion.

In Time based, files are deleted securely after a specified duration. The methodology is to encrypt file with data key and the data key is again encrypted with a control key and the control key is preserved by a key manager. The control key has time associated to it; therefore it will be removed automatically by the key manager after the time expires. The data key, cannot be generated without control key therefore the file is inaccessible. Policy based deletion, is associated with file access policy and policy is in turn associated with a control key and all the control keys are maintained by the key

Revised Manuscript Received on December 22, 2018.

K. Pradeep kumar, Pydah College of Engg.& Tech. Visakhapatnam, AP, India (Email: kpk0208@gmail.com)

Dr. G. Lakshmeeswari, GITAM Deemed to be University, Visakhapatnam, AP, India(Email: lakshmeeswari.gondi @gitam.edu)

manager. The idea is that, the file is encrypted with data key and the data key is further encrypted with a control key associated with the policy. when the policy is revoked the key is removed by the key manager making the file access impossible.

The limitation with FADE is that file deletion is based on policy revocation and it involves key manager who has to be regarded as trusted. The concern is compromising of the key manager[4].

SFADE proposes key manager removal so the cloud servers and users are in direct contact with each other. Depending upon the policy, FADE will generate a key that will be used to encrypt data. Using the same policy, adjunct key is created. This adjunct key is used to encrypt the first key. The encrypted file and the first key are stored to the cloud. The generation of key depends on policy. If the policy is revoked, all encrypted files are also revoked. The limitation of SFADE is that it does not have file sharing feature in a secure way[6].

SFADE+ proposes file sharing feature in a secure way. Encrypts the data key before sharing it. The encrypted data keys reside in a storage which is accessible to both the data receiver and the data owner. Here also the generation of key depends on policy and revoking the policy leads to revocation problem. The limitation of SFADE+ is that the generation of key depends on policy. If the policy is revoked, all files encrypted will be unreadable leading to revocation problem.As per NIST (National Institute of Standards and Technology) guidelines, "Encryption is not generally accepted means of sanitation"

Even after all these approaches for deleting data, it was found that retrieve deleted files. As per the security study of Amazon EC2 service. They could recover a minimum of 6 to a maximum of 40,000 files per a Amazon machine Instance(AMI)[16]. Even though the user is the owner of his data, it is the CSP who has physical custody of his data that means, the provider's policy actually governs the fate of data.

Even though the file is marked as deleted, the contents of the file are not wiped off the disk. The data can be retrieved if intended using recuva, puran file recovery, stellar data recovery tools. When a user's SLA expires all files in his allotted storage space are shifted to retention disk and (File Allocation Table)FAT is emptied so as to allot it to a new user with traces of residual data in the data block. There is every possibility for the new user recover and exploit that data by recovering it.

3. PROPOSED METHODOLOGY

On deleting a file, all the clusters occupied by the file are added to the free space available on disk by the OS. So, the sectors that a file previously occupied are now available for storing new files. If the sectors are not reused or reallocated to new files by the OS, there is every possibility of recovering the files by using recovery tools. The major limitation with most of recovery tools is their inability to retrieve data if its overwritten. This limitation is exploited in the proposed methodology.

Overwriting data to a specific portion of disk is only made possible by replacing the contents of the old file with new one before deletion. Once the data is overwritten, the previous contents may not be easily retrieved. Fig. 1 details the proposed architecture for deleting a file such that its contents would not be recoverable after deletion.

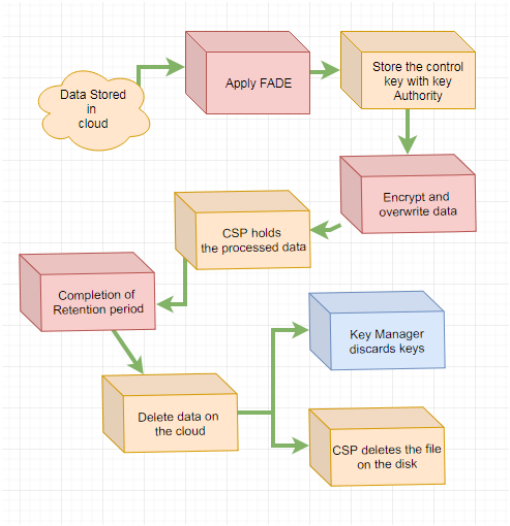

Fig. 1 : Proposed architecture

3.1 User functionalities

A user never wants his data to be disclosed to an anonymous party. The anonymous party may be an ordinary individual, hacker, business organization or an attacker. Inorder to prevent this disclosure, the user would execute the Encrypt-Overwrite Process(EOP), on the verge of expiry of SLA or when the user wants to recede from cloud usage. The execution of EOP by user ensures that the data is overwritten multiple times. Therefore, the probability of recovery of original data using recovery tools is minimized when the same physical space is allotted to another user on current user termination.

Depending on the sensitivity of data user can decide on the number of times he want to overwrite specific data during EOP.

Encrypt-Overwrite Process:

Step 1: Generate hash value of the file.
Step 2: Encrypt the file using the hash valueas key.
Step 3: The output of Step 2 is written back to the Same file.
Step 4: Repeat step 1, 2 and 3 for 'n' number of Times.

3.2 CSP's functionalities:

When a user parts with the cloud, his data is moved on to separate storage location. CSP has to retain the data for a certain period of time, this time period is called as retention period. The retention period various from one geographical location to other and is governed by the law of the there government. It is the duty of the CSP to delete the data after the retention period. FADE assures a secured storage and deletion of data.

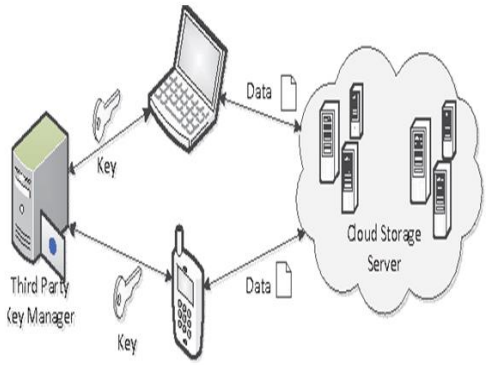

Fig. 2 : FADE architecture

The ideology of this algorithm is to encrypt file with data key and next the data key is encrypted using control key and this key is stored with key authority. Key authority is a trusted third party. With expiry of retention period control key is removed by the key authority. The data as well as data key are inaccessible without control key.

Encrypting the data and discarding the keys after the retention period makes the data unrecoverable, but any compromise by the key authority, would help un-authorized recovery of data. In order to limit this the CSP performs the EOP after the FADE is processed. This process helps in:

- 2-step data security
  - Compromising only the key holding authority would not disclose the data, as decryption of data using control keys would not give the actual content.
  - Compromising only CSP would result in an encrypted data after reverting the EOP. This encrypted data can be decrypted only with the keys held by key authority.
- Deleting data after retention period assures that:
  - The entry in file system is deleted
  - Key authority discards the key values.
  - CSP discards 'n' value.
  - Due to multiple overwrites on the same area in the data block. The probability of recovering the original data is minimalistic.

The functionalities performed by the CSP assures that the data deleted after retention period is also permanently deleted. When user and CSP execute their functionalities, complete and permanent deletion of data is assured.

4. RESULTS

Fig. 3 shows the percentage of original contents of the file that have been retrieved after overwriting it for 5 times, with different data each time.

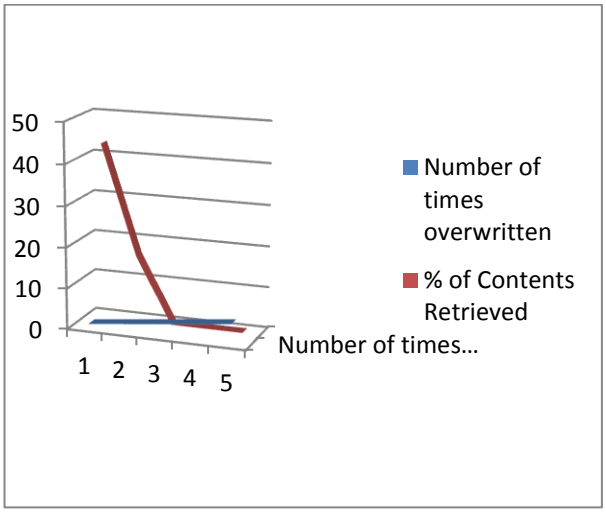

Fig. 3: Chart depicting the % of old content retrieved from overwritten file.

5. CONCLUSION

Deletion of data from physical storage will only remove the pointer to the file but the data may be present until it is overwritten. One way of preventing data recovery of deleted data is to overwrite the contents of file.

REFERENCES

1. Scott Wolchok, Owen S Hofmann, Nadia Heninger, Edward W Felten, J Alex Halderman, Christopher J Rossbach, Brent Waters, Emnet Mitchel" Defeating Vanish with low-cost Sybil Attacks Against large DHTs", conference proceedings of the Network and Distributed System Security Symposium, NDSS, SanDiego,california, USA, 2010
2. Lingfangzeng, Zhan shi, Shengjie Xu, Dan Feng "Safe vanish: An Improved data self destruction for protecting data privacy" ,IEEE International Conference on Cloud Computing Technology and Science, 2010.
3. Yang Tang, Patrick P. C. Lee, John C. S. Lui, and RadiaPerlman, "FADE: Secure Overlay Cloud Storage with File Assured Deletion", IEEE, 2005.
4. R. Perlman. "File System Design with Assured Delete". In ISOC NDSS, 2007.
5. AshfiaBinte Habib, Tasnim Khanam, Rajesh Palit "Simplified File Assured Deletion (SFADE) - A UserFriendly Overlay Approach for Data Security in Cloud Storage System" , IEEE,2013.
6. Raisa Nusrat, Rajesh Palit "Simplified FADE with Sharing Feature (SFADE+): A Overlay Approach for Cloud Storage System",,IEEE,2017.
7. JinboXiong ,Zhiqiangyao , Jinfeng Ma, Ximeng Liu , QI Li "A Secure Document Self-destruction Scheme: An ABE Approach",IEEE,2013.
8. JinboXiong ,Zhiqiangyao , Jianfeng Ma, Ximeng Liu, QI LI "A Secure Document Self- destruction Scheme with Identity Based Encryption",IEEE,2013
9. FU Xiao, WANG Zhi-jian ,WU Hao, YANG Jia-qi,WANG Zi-Zhao "How to send a Self-destructing Email",IEEE,2014.
10. R.Barona , E.A.Mary Anita "A Survey on Data Breach Challenges in Cloud Computing Security : Issues and Threats", IEEE,2017.

11. Jayashree Agarkhed ,Ashalatha R, "An Efficient Auditing Scheme for Data Storage Security in Cloud" ,IEEE,2017
12. LingfangZeng,Zhan Shi*, Shengjie Xu, Dan Feng "Safe Vanish: An Improved Data Self-Destruction for Protecting Data Privacy", IEEE,2010.
13. Igarramenzakaria, Hedaboumustaha "FADETPM: Novel Approach of File Assured Deletion based on Trusted platform module",IEEE,2017.
14. Yuanyuan Zhang, JinboXiong, Xuan Li, Biao Jin, Suping Li, Xu An Wang "A Multi-Replica Associated Deleting Scheme in Cloud", 10th International Conference on Complex, Intelligent, and Software Intensive Systems, 2016.
15. Marco Balduzzi, JonasZaddach, Davide Balzoro, Enginkirda, Serigo Loureiro," Asecurity analysis of Amazon's Elastic compute cloud service" ACM,2012.