ISSN:2231-0711

Sure Mamatha et al | IJCSET(www.ijcset.net) | February 2016 | Vol 6, Issue 2, 95-97

A Survey on Different Techniques for Assured File Deletion in Cloud

Sure Mamatha[#1], V.Lakshma Reddy[*2], A.Amruthavalli[*3]

[1,2,3]*Assistant Professor, Department of CSE,*
PACE Institute of Technology & Sciences,
Ongole, Andhra Pradesh

Abstract-Cloud computing has been one of the most prominent buzzwords of the IT industry due to its revolutionary model of computing as utility. Cloud computing promises increased flexibility scalability and reliability with promising decreased operational and maintenance costs as every organization is accumulating tons and tons of data, it is quite difficult for the organizations to manage that huge amount of data, So instead of self-maintaining the data the organizations can outsource their data to the cloud storage and can reduce the data management overhead. Cloud storage offers an abstraction of infinite storage space for clients to outsource their data and access it in a pay-as-you-use manner. While we are outsourcing the data to the cloud storage services, managed by third parties, several security concerns will arise in terms of privacy and integrity of the data .The traditional and cloud storage are becoming highly reliable in recovering the data from a disaster/ failure. For them to be reliable the cloud service providers are creating multiple redundant copies of the data and they were spread through the cloud for reliability and availability, without the knowledge of the data owner. One specific issue is as that many number of copies of the data is created, it is hard to delete all those copies, the cloud service may forget to remove all the copies / intentionally may not delete. Some of the copies of data upon request of deleting the data by the data owner. So to ensure the privacy and integrity of data outsourced to the cloud storage, we must design a practically implementable and readily deployable application which ensures security of the outsourced data and provides the data owner the assurance that the data was in a safe state, and all the copies of the data were deleted .this paper discusses several mechanisms available to ensure the assured file deletion. First we discuss the advantages and disadvantages of various mechanisms, and then we propose a practically implementable, readily deployable mechanism which works on top of the existing cloud architecture with minimal management overhead.

Key words: cloud storage, policy based file assured deletion, time based deletion

1. INTRODUCTION

Now a day the cloud has become the vehicle for delivering resources such as computing and storage to customers in a pay as you use manner. With the rapid growth of day to day data, every organization wants to reduce the data storage& management cost. [1]Cloud storage is a model of networked enterprise storage where data is stored in virtualized pools of storage which are generally hosted by third parties. Hosting companies operate large data centres, and people who require their data to be hosted buy or lease storage capacity from them. However as data is maintained by third party cloud service provides several security concerns will arise in terms of the privacy and integrity of data. One of the straightforward approach to ensure privacy of the data is to apply some cryptographic techniques with some encryption keys

Figure 1.Cloud Storage

2. SURVEY MECHANISMS:

With the user's increasing reliance on the web services for everything, personal data was cached, copied and archived by the third parties without the knowledge and control of the user. The privacy and integrity of the archived and copied, cached data must be ensured for the data owner.in order to securely delete the data many techniques have been proposed, which in general overwrite the data in order to delete the data, however techniques to overwrite the data heavily depend on the properties of the physical storage in which data was stored, but with cloud computing providing virtualized storage models, physical control over data storage locations is no longer possible.

(i)Time based file assured deletion:

Time based file assured deletion was first introduced in [2], according to this the files can be securely deleted and remain in accessible after the pre-defined time. In this the file is encrypted with a data key, and this key is further encrypted with a control key that is maintained by a separate key manager. The control key is time-based i.e. it will be completely removed by the key manager upon the expiration of the time, the expiration time will be specified when the file is created. Without the control key, the data key and the data file will remain encrypted and inaccessible. But the drawback of time based assured deletion is , the files become inaccessible ,after the

expiration of time, but the file in an encrypted form will be there , if an intruder figure out the key , he can try to recover the data and he may succeed too.

(ii)Vanish:

Vanish guarantees that all the copies of certain file become inaccessible after expiration of user specified time, even if an attacker obtains the copy of the file and key, with which the file was encrypted .vanish guarantees this with the novel integration of cryptographic techniques with global scale , P2P, distributed hash tables(DHTs).Vanish encrypts the data locally with a random key K , not known to the data owner, removes the local copy of the key and distributes the bits of the key across random indices in the DHT. The effectiveness of the vanish depends on the properties of the DHTs.vanish takes a data object X and encapsulates it into a VDO V, to encapsulate the data object X vanish encrypts the data locally with a random key K, not known to the data owner, removes the local copy of the key and distributes the bits of the key across random indices. Vanish uses threshold secret sharing to split the data key K into N pieces .The threshold determines how many of the N sharers are needed to regenerate the original Key K.The Vuze-based system can support8-hour timeouts in the basic Vanish usage modeland the openDHT-based system can support timeoutsup to one week.

Vanish was available as a Firefox plugin for the Gmail service that provides the opportunity to send and read self-destructing emails. Vanish does provide the security against the store sniffing attack [3] .but vanish does not ensure the protection against the Sybil attack [4].

(iii)**FADE**:

Both vanish and time based file assured deletion focuses only the assured file deletion upon expiration of time, but to give more fine grained control to the data owner a policy based file assured deletion (FADE) mechanism was proposed. FADE is a practically implementable and readily deployable cloud storage system that focuses on assured file deletion upon requests of delete, with policy-based file assured deletion. Policy-based file assured deletion is developed upon conventional cryptographic techniques, i.e. it encrypts the outsourced data files to guarantee their confidentiality and integrity, assuredly deletes the file to make them unrecoverable upon request of delete with revocation of file access policies. FADE divides the management of encrypted data and management of encryption keys. The encrypted data remains on the untrusted cloud storage, while encryption keys were maintained by a separate key manager which follows a quorum scheme [5].

In FADE, each file is associated with a single atomic file access policy or a Boolean combination of atomic policies. Each policy is associated with a control key, and all the control keys are maintained by a separate key manager. The data file is encrypted with a data key, and the data key is further encrypted with a control key corresponding to the policy combinations. When a policy isrevoked, the corresponding control key will be removed from the key manager, with this the data key and hence the encrypted data file cannot be recovered. The main idea of assured file deletion is to make the file inaccessible upon request of delete, even though it was not physically removed from the cloud storage. The FADE contains three categories of participants: data owner, key manager, cloud storage.

Data owner: The data owner is the entity, whose data is to be stored in the cloud storage, the data owner can be a mobile device, a user level program or it may be a file system in a PC. The data owner encrypts the data to be outsourced to the cloud, interacts with the key manager to perform the necessary cryptographic operations.

Key Manager: The key manager maintains the keys according to the policies, which are used to encrypt the data. It performs the operations like encryption, decryption, revocation and renewal of control keys.

Cloud storage: the cloud storage was maintained by a third party cloud service providers like Amazon S3 and keeps data on behalf of owner of the data. FADE guarantees that we do not require any protocol or implementation changes to the cloud storage to support this, because the storage clouds were owned and managed by third party cloud service providers, so there should not be any structural changes in the way it was implemented. But FADE does not ensure version control, which is very much important as the need for cloud storage is increasing, in order to eliminate the redundant copies of the data. FADE does not ensure the access control to the genuine data owners. Access control to the outsourced data was provided in [6], in this in order to ensure the access control, cipher text-policy attribute Based Encryption [7] can be used. In cipher text –policy attribute based encryption the user will request the key manager to decrypt the file by supplying authentication credentials to show that he indeed satisfies the policies associated with the file. In this each user will gets a cipher text policy attribute based encryption –based private access key that depends on a set of attributes that the user satisfies , then this private access key is distributed to the users who satisfies the corresponding policies . The key manager will maintain the cipher text – policy attribute based encryption public access key and uses it to encrypt the message responses returned to the users. In this way having two sets of keys will ensure both access control and assured deletion to the data owners.

FADE Version: with the increasing use of mobile devices the demand, need for cloud storage is also increasing day by day, because a mobile device does not come up with bigger memory. So the cloud storage is becoming attractive, but the major challenge is to provide assured file deletion, version control, access control.

As every organization is accumulating tons and tons of data, the demand for cloud backup storage is increasing, otherwise the data management overhead increases for the organizations. One important application of cloud backup systems is version control. FADE does provide assured deletion but it does not guarantee the version control, which is very much important to eliminate the redundant copies of the data. FADE Version [8] is a secure cloud backup system that supports assured file deletion and Version control. Fade Version allows a fine grained assured deletion, i.e. it allows the cloud users to specify which version of the file to be assuredly deleted. In order to

eliminate the redundant copies of the data Fade Version follows the notation of Deduplication [9].

There are many commercial cloud backup systems available in the market, like Dropbox [10], nasuni [11], jungle Disk [12] and open source like Cumulus system [13], which considers a thin cloud interface, i.e. it only offers the basic features like put, get, list and delete. It divides the data files into chunks, only modified chunks will be uploaded to the cloud, but cumulus does not ensure assured deletion. All these will provide the version control and store different versions of backups. But the existing cloud assured deletion mechanisms like vanish and Fade are incompatible with the existing version controlled systems such as cumulus, nasuni.

REFERENCES

1. http://en.wikipedia.org/wiki/Cloud_storage
2. R. Perlman. File System Design with Assured Delete. In ISOC NDSS, 2007.
3. www.packetwatch.net/documents/papers/layer2sniffing.pdf
4. www.few.vu.nl/~mconti/teaching/ATCNS2010/ATCS/Sybil/Sybil.pdf
5. A. Shamir. How to Share a Secret.CACM, 22(11):612–613, Nov 1979
6. http://www.cse.cuhk.edu.hk/~pclee/www/pubs/tdsc12_tech.pdf
7. www.cs.utexas.edu/~bwaters/publications/papers/cp-abe.pdf
8. www.cse.cuhk.edu.hk/~pclee/www/pubs/cloudsec11.pdf
9. http://plan9.bell-labs.com/sys/doc/venti/venti.pdf
10. Dropbox. http://www.dropbox.com, 2010.
11. Nasuni. http://www.nasuni.com
12. JungleDisk. http://www.jungledisk.com/, 2010.
13. http://www.cs.cornell.edu/courses/cs6464/2009sp/lectures/02-cumulus.pdf