ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

Using Storage Framework a Self-Vanishing of Data System in the Cloud

Rekha. T

P.G. Student, M.Tech, 4th Sem, Department of Computer and Engineering, VTU CPGSB, Visvesvaraya Institute of Advanced Technology (VIAT), Muddenahalli, Chickaballapur Dist, Karnataka, India

ABSTRACT: Large amount of data can be stored in the cloud for longer period of time along with the security. In the cloud many users may have their data which is highly confidential. Even though user data may be secured by account numbers, passwords, notes, pin numbers and other important information, there may be chance of using or misusing of data by a miscreant. To maintain the good consistency & precision the cloud service providers (CSPs) replicates the data geographically without the authorized users permission. Self-vanishing data is mainly aims to protecting the users privacy of data or users secured data. User specified time interval to vanish the data whenever he does not required his data to be stored in the cloud, then all the data and their copies become vanished without any user intermediate & it becomes unreadable. In addition to this, the decryption key is also vanished after the user specified time interval. So there will be no longer data has been stored in cloud only if user not required his data to be stored in the cloud. This paper is using cryptographic techniques and storage framework to solve above challenges.

KEYWORDS: Cloud Computing, Self-Vanishing Data, Storage Framework, Privacy Of Data, DHT.

I. INTRODUCTION

Cloud computing plays a major role for storing the large amount of data for organization or the individuals. Cloud provides the data storage and also provides the services like Platform-as-a-service(Paas), Software-as-a-service(Saas), Infrastructure-as-a-service(Iaas). Because of these services the organization and individuals. These are mainly focusing towards cloud.

Cloud provides the mobility since it is internet based technology, users are more interested in storing and accessing their personal data very easily. The users data may contain passwords, pin numbers, account numbers and some more important information to provide the security of personal data. All the files can be maintained in a single directory in cloud instead of storing the individual files. The user specifies the time interval for each of its confidential data i.e., file that are stored in the cloud. Every personal data that are maintained as confidential file stored in the cloud may not be required for the longer period of time by the user. After the user specified time interval expired the files and all the replicas will be self-vanished from the cloud.

Vanish methodology was proposed to delete the file after specified time interval. In this methodology the secret key is important. Here secret key will be divided and stored in the distributed hash table (DHT), DHT is one of the characteristic of P2P, the node. For every 8 hours this DHT will be getting refreshed. So the keys present in the node will be deleted, because of this decryption of file becomes difficult as the user may not get the enough number of keys to do it.

Uncontrolled in how long the key can survive is one of the disadvantages for Vanish. That is in Vanish methodology key cannot survive for longer period of time. To overcome this challenge this paper is proposed which is dependent on Active Storage Framework. The Self Vanishing Database system mainly stipulate two modules, one is self-vanish method object that is fraternized with each and every secret key and another is for each secret key the survival time parameter.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

Self Vanishing Database system offers:

1) To store the clients distributed key in the object storage system Key distribution algorithm is used which is based on Shamir's algorithm is mainly used as core algorithm. Here the object based storage interface is used to store and manage the keys which are divided and stored in the DHT that are based on Active Storage Framework.
2) Self Vanishing Database systems are supported to store the all deleting files and random encryption keys that is stored in secondary storage

II. RELATED WORK

Levy et al. (2009) proposed "Vanish: Increasing Data Privacy with Self-Destructing Data" [2]:
Without to our knowledge or control of Third Parties are used to cached, copied, and archived the user's Personal data. Once user specified time interval we like to ensure that all copies of data will be vanished and become unreadable, without any specific action on the part of a user, and even if an unauthorized user(attackers) obtains both a cached copy of that data and the user's cryptographic keys and passwords.
Vanish overcomes the above challenges With the help of novel integration of cryptographic techniques. The goal is to self-vanish the data automatically after it is no longer useful to the user. Vanish system leverage the services provided by decentralized, global-scale P2P infrastructures and, in particular, Distributed Hash Tables (DHTs). DHTs are designed to implement a robust index-value database on a collection of P2P nodes. With a random encryption key not known to the user will Vanish encrypts a user's data locally, & destroys the all local copy of the key, and then sprinkles bits (Shamir secret shares) of the key across random indices (thus random nodes) in the DHT.

Vanish architecture:
Data object D is taken by Vanish in order to encapsulate it into VDO. To encapsulate the data *D*, Vanish picks a random data key, *K*, and encrypts *D* with *K* to obtain a cipher text C.

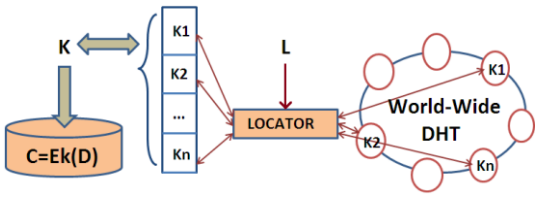

Figure 1: Vanish System Architecture

Figure shows how to split the data key K into N pieces $K_1 \ldots K_N$ vanish uses threshold secret sharing. The application or the user can set the threshold, which is the parameter of secret sharing. To reconstruct the original key, N shares are required which is determined by threshold.

Advantages:
Vanish focus on the post-facto, retroactive attacks; that is an old or forgotten or unreachable copies of data will be defends the user against future attacks on it. The attacker's must develop an infrastructure capable of attacking all users at all times. Therefore attacker's job is very difficult. It does not require special security hardware resources, or special operations on the part of the user. Since Solution utilizes existing, popular, researched technology used since 2001. Data definitely destroyed since DHT utilizes half-life (churn) of nodes.

Disadvantages:
Universally this mechanism is not applicable to all users or all data types. They mainly focusing in particularly on sensitive data that before falling into the wrong hands a user would wish to see destroyed data early. Vanish applications may compose VDOs with traditional encryption systems like PGP and GPG. It does not defend against denial of service attacks that could prevent reading of the data during its lifetime. In this case, the user will naturally need to manipulate the PGP/GPG keys and passphrases.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

Tang et al. (2010) proposed "FADE: A secure overlay cloud storage system with File Assured Deletion" [3]:
Data may be unexpectedly disclosed in the future due to malicious attacks on the cloud, so storing data permanently is undesirable, or careless management of cloud operators leads to misuse of data by the attackers. The challenge of achieving assured deletion is that we have to trust cloud storage providers to actually delete data, but they may be reluctant in doing so. Also, multiple backup copies of data will be typically present in cloud storage providers for fault-tolerance reasons. It is uncertain, from cloud clients' perspectives, whether cloud providers reliably remove all backup copies upon requests of deletion.

FADE is a secure overlay cloud storage system that provides fine-grained access control and assured deletion for outsourced data on the cloud, while working seamlessly atop today's cloud storage services. In FADE, active data files that remain on the cloud are associated with a set of user-defined file access policies (e.g., time expiration, read/write permissions of authorized users), such that data files are accessible only to users who satisfy the file access policies. In addition, FADE generalizes time-based file assured deletion (i.e., data files are assuredly deleted upon time expiration) into a more fine-grained approach called policy-based file assured deletion, in which data files are assuredly deleted when the associated file access policies are revoked and become obsolete.

The FADE system:
The FADE system is composed of two main entities: FADE clients and Key managers.
• **FADE clients.** A FADE client (or client for short) is an interface that bridges the data source (e.g., file system) and the cloud. Depending on data it applies encryption or decryption to the outsourced data files that may be uploaded to or downloaded from the cloud. To perform the necessary cryptographic key operations it also interacts with the key managers

• **Key managers**. FADE is developed by minimum group of key managers together. Based on FADE & how keys are maintained helps for assured deletion and access control policy, each of which in turn is a standalone entity.

Advantages:
FADE decouples the management of encrypted data and encryption keys, such that encrypted data remains on third-party (untrusted) cloud storage providers, while encryption keys are independently maintained by a key manager service, whose trustworthiness can be enforced using a quorum scheme. FADE generalizes time-based file assured deletion into a more fine-grained approach called policy based file assured deletion, in which files are associated with more flexible file access policies and are assuredly deleted when the associated file access policies are revoked and become obsolete.

Disadvantages:
Batch files operations and block update operation will not be supported. It has to transmit metadata along with the file.

III. SYSTEM ARCHITECTURE

The architecture shows the Self Vanishing Database system in figure 2. It consists of three components. They are mainly based on Active Storage Framework. They are:

i. Metadata server.
ii. Application node.
iii. Storage node.

Metadata server: consists of many managements like user, session, server, key and file management.
Application node: is used by the user for the services like storage in the Self Vanishing Database system.
Storage node: it mainly consists of two subsystems, one is storage subsystem like <key, value> and another is runtime subsystem like Active Storage Object. These subsystems have many features like: Store subsystem is dependent on Object Storage Component which performs the operations like managing objects that are stored in storage node. The

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2015

object is represented uniquely, called as objectID, which is used as key. Active Storage Object runtime subsystem is dependent on Active Storage Agent Model.

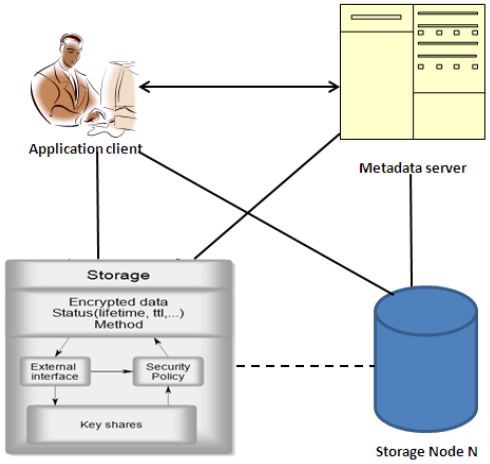

Figure 2: Architecture of Self Vanishing Database system

Active Storage Object: The time-to-live (TTL) value property has active storage object that are defined by the user object. Users will give some value to their data stored in the cloud. That value is called as TTL value. TTL value is defined as its value that is specified by the user to his private files. Time-to-live is the survival time for file that specifies how long the file exists in the cloud. Files will be self-Vanished once the TTL value expires.

IV. CONCLUSION

The cloud computing is mainly concerned with the data security and privacy of data. This paper mainly surveyed on the privacy of data for sensitive files that are stored in the cloud. Here we are mainly concentrating on the Self Vanishing Database system methodologies that mainly used for privacy & security of data. In Self Vanishing architecture it consists of three components such as application node, metadata server and storage node. By using few of the active storage and cryptographic techniques, Self Vanishing Database system provides security & privacy for confidential files.

REFERENCES

[1] Lingfang Zeng , Shibin Chen , Qingsong Wei , and Dan Feng, "SeDas: A Self- Destructing Data System Based on Active Storage Framework," IEEE Transactions On Magnetics, Vol. 49, No. 6, June 2013.
[2] Levy et al. (2009) proposed "Vanish: Increasing Data Privacy with Self-Destructing Data".
[3] Tang et al. (2010) proposed "FADE: A secure overlay cloud storage system with File Assured Deletion".
[4] Shruthi, Ramya N, Swathi S.M, Sreelatha P.K (2015):"Self-Vanishing of Data in the Cloud Using Intelligent Storage"
[5] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[6] R. Geambasu, T. Kohno, A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," in Proc. USENIX Security Symp., Montreal, Canada, Aug. 2009, pp. 299–315