SEDAS - Self Destruction Data System

Maruthavani E
Computer Science And Engineering Karpagam
Institute Of Technology Coimbatore,
India

Chandra Prabha T
Computer Science And Engineering Karpagam
Institute Of Technology Coimbatore,
India

Sakthivel Murugan T
Computer Science And Engineering Karpagam
Institute Of Technology Coimbatore,
India

Hemalatha M
Computer Science And Engineering Karpagam
Institute Of Technology Coimbatore,
India

Manikandan D
Computer Science And Engineering Karpagam
Institute Of Technology Coimbatore,
India

***Abstract*- On Cloud personal data is stored that may contain account numbers, passwords, notes of users, and these data and other information that could be important for the user may be misused by a hacker, a competitor, or by a court of law. These data which we store on cloud are cached and copied by Cloud Service Providers; this often happens without users' authorization and control. Self-destruction of data mainly aims at protecting the data's privacy of user. Without any information to user there data and copies get destructed after the user specified time. Also the decryption key gets destructed after the user-specified time. Various cryptographic techniques along with active storage techniques are used in this system to meet the challenges which are faced by the user. We implemented a proof-of-concept self destructive prototype. The results**
demonstrate that the system can be practically used and meets all the goals for preserving privacy by the self destructive prototype. The throughput for uploading the data and downloading the data with the proposed system acceptably decreases,
while the latency for upload/download operations with the self-destructing data mechanism increases if we compare to the system without self destruction data mechanism.

Index Terms- Cloud computing, Active storage data privacy, self-destructing, data privacy.

I. INTRODUCTION

As Cloud computing and mobile Internet are getting more and more popularized, In people's life cloud services are becoming important because of this. People are now and then requested to submit or post their personal private information through the internet on the cloud. When people post their data, on the cloud they hope that the security will be provided by the service providers to protect their data from leaking, so that invading of privacy will not be done by other people. Security of their privacy is on more risks as people rely more on internet and cloud technology. The systems or network must cache or copy the data that is being processed, transformed and stored. Because these copies are essential for systems and the network. As people have no knowledge about these copies which they are putting on cloud or internet and cannot control them, their privacy of these copies may get leak. On the other hand, Cloud Service Providers negligence, hackers' intrusion or some legal actions by all this also privacy can be leaked. To protect people's privacy all these problems present formidable challenges.

On Cloud personal data is stored that may contain account numbers, passwords, notes of users, and this and other information that could be important to the user can be misused by a hacker, a competitor, or by a court of law. These data which we store on cloud are cached and copied by Cloud Service Providers; this often happens without users' authorization and control. Self-destruction of data mainly aims at protecting the data's privacy of user. Without any information to user there data and copies get destructed after the user specified time. Also the decryption key gets destructed after the user-specified time.

II. OBJECTIVES

Objectives of Proposed System to implement a self destructing data system are as follows:- Two modules have been defined by the self destructive system, first one is a self-destruct method object with which each secret key part is associated and second one is that with, each secret key part has a survival time parameter. In this case, the system can be used as a

general object storage system and the requirement of self destructing data with the control on the survival time can be meet all this.

Our objectives are summarized as follows.

1) Our focus is on the key distribution algorithm, Shamir's algorithm, which is the key distribution algorithm, is the core algorithm; to make the Object storage system implementation of client distributing is done To implement a safety destruction of data with equal divided key these methods has been used.

2) An object-based storage interface is used based on the active storage framework for storing and managing the equally divided key.

3) System is practical to use and fulfills all the requirements of privacy-preserving of the data this can be seen through the functionality and security properties and the evaluation of this prototype. Reasonably low runtime overhead is imposed by the prototype system.

4) System supports security erasing files and random encryption keys stored in a hard disk drive or solid state drive, respectively.

5) Using load balancing and round robin algorithm for managing the load on the nodes.

III. RELATED WORK

A. Existing System:

New ideas for sharing and protecting privacy is supplied by a pioneering study of vanish. In the Vanish system, the secret key provided is divided and stored in a point to point system in with distributed hash tables.

The system can maintain secret keys with the joining and exciting of point to point network. The characteristics of point to point are that, the distributed hash tables will refresh every node after every eight hours. While the characteristics of Shamir Secret Sharing Algorithm is that when we will not get enough parts of a key, data cannot be decrypt which was encrypted with this key, which means the data cannot be recovered as the keys have been destroyed. Some special attacks to characteristics of point to point are challenges of Vanish, uncontrolled in how long the key can survive.

Vanish is a system used for creating messages that automatically self-destruct after a period of time. It integrates cryptographic techniques with global-scale, point to point distributed hash tables. Distributed hash tables have the property to discard data older than a certain age. After the data expiration time the key gets permanently lost and because of this the encrypted cannot be read permanently. In Vanish system each message is encrypted with a random key and the share of the key is stored in a large, distributed hash tables which is public.

However, Sybil attacks may compromise the system by continuously crawling the distributed hash tables and saves each stored value before it get lapse out and comparison with the mentioned estimated the total cost is less. More than 99% of Vanish messages can be efficiently recover keys in this.

B. Proposed System:

The self destructive system defines two new modules, a self-destruct method object that is associated with each secret key part and each secret key part has its own survival time parameter. The system can be used as a general object storage system and with controllable survival time the self destructive system can meet the requirement of self destruction

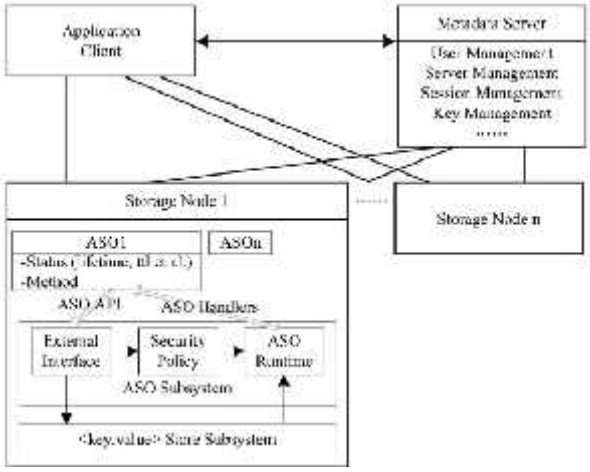

Fig 1 Self destructive system architecture

A. Self destructive Architecture

Fig. 1 shows the architecture of self destructive. There are three parties based on the active storage framework.

i) *Metadata server:* It is responsible for user management, server management, session management and file metadata management.

ii) *Application node:* The application node is a client to use storage service of the self destructive.

iii) *Storage node:* Each storage node is an OSD. It contains two core subsystems: key value store subsystem and active storage object runtime subsystem. The key value store subsystem which is based on the object storage component and is used for managing objects stored in storage node: lookup object, read/write object and so on. The object ID is used as a key. The associated data and attribute of the node are stored as

values. The active storage object runtime subsystem based on the active storage agent module in the object-based storage system this system is used to process the active storage request which is send from the users and along with this it manages the method objects and policy objects also.

B. Active Storage Object

An active storage object derives from a user object and has a time-to-live value property. The time-to-live value is used to trigger the self-destruct operation. The time-to-live value of a user object has the property infinite so the user object will not be deleted until a user deletes it manually. On the other hand the time-to-live value of an active storage object is limited so an active object will be deleted when the value of the associated Policy object is true.

C. Self-Destruct Method Object

A self-destruct method object is a service method. It needs three arguments. The lun argument specifies the device; the pid argument specifies the partition and the obj_id argument specifies the object to be destructed.

D. Data Process

To use the self destructive system, user's applications should implement logic of data process and act as a client node. There are two different logics: uploading and downloading.

i)*Uploading file process:* When a user wants to upload a file to the storage system and along with that also wants to store his key in this System, he has to specify the file, the key and time-to-live as arguments for the uploading procedure. Fig. 3 presents its pseudo-code. In these codes, we assume data and key has been read from the file. The ENCRYPT procedure uses a common encrypt algorithm or user-defined encrypt algorithm. After uploading data to storage server, the key shares that are generated by Shamir Secret Sharing algorithm are used to create active storage object in storage node in the Self destructive system. ***ii)***Downloading file process: Any user who has relevant permission can download data stored in the data storage system. The data must be decrypted before use.

E.Data Security Erasing in Disk

We must secure delete sensitive data and reduce the negative impact of OSD performance due to deleting operation. The proportion of required secure deletion of all the files is not great, so if these parts of the file update operation changes, then the OSD performance will be impacted greatly.

Our implementation method is as follows:

i)The system pre specifies a directory in a special area to store sensitive files.

ii) Monitor the file allocation table and acquire and maintain a list of all sensitive documents, the logical block address.

iii) Logical block address list of sensitive documents appear to increase or decrease, the update is sent to the OSD.

iv)OSD internal synchronization maintains the list of logical block address, the logical block address data in the list updates.

IV. PROPOSED PLAN OF WORK

1. Development of Login System:-

In this module we have made a login page, where a new user has to first register itself my filling the given fields after he submits its information, he will receive a mall on his email Id that by clicking on the given link he can login to the SeDas account. As he clicks on the link his account becomes active and he can use the services of SeDas.

2.Development of Active Storage Framework:-

An active storage object that is derived from a user object and has the time-to-live value property. The time-to-live value is used to trigger the self-destruct operation. The time-to-live value of a user object is infinite i.e. user object will not be deleted until a user deletes it manually. The time-to-live value of an active storage object is limited so an active object will be deleted when the value of the associated Policy object is true. This TTL we will use at the time of uploading the file for encryption, at that time we will specify the time period in sec for the file.

3.Development of login tracking of the user:--

To use the Self destructive system, user's applications should implement logic of data process and act as a client node. There are two different logics: uploading and downloading.

i)*Uploading file process:* When a user wants to upload a file to a storage system and along with it also wants to store his key in this System, he should specify the file, the key and time-to-live as arguments for the uploading procedure. We assume data and key has been read from the file. The ENCRYPT procedure uses AES encrypt algorithm. After uploading data to storage server, key shares generated by Shamir Secret Sharing algorithm will be used to create active storage object in storage node in the self destructive system. *ii)Downloading file process:* Any user who has relevant permission can download data stored in the data storage system. The data must be decrypted before use. The whole logic is implemented in code of user's application.

4.Transfer of Encrypted file between users:-

In this module we are specifying the user list which is using the SeDas system; in this user list we have attributes as name of user, gender and his email address. If a user wants to share his information with other user he can mail the encrypted file path to the

other user by using his ID specified in user list. The other user can download the file by copying the path to its download frame and if that file TTL property has not been expired the file can be downloaded by the user.

5.Development of deletion module in case user logs out:-

In this module we are showing all the downloaded data of the user along with its object ID, if the user before log out thinks that he will not require this data further so he can delete the data from its database. For this we have provided a delete button in front of every downloaded data, so user can click on it and the data can get deleted from user downloads

6.Testing and optimization of our system:-

We input the full path of file, key file, and the life time for key parts. The system encrypts data and uploads encrypted data. The life time of key parts is 150 s for a sample text file with 101 bytes. System prompts creating active object are successful afterwards and that means the uploading file gets completed. The time output finally is the time to create active object. Self destructive was checked and corresponded with changes on work directory of the storage node. The sample text file also was downloaded or shared successfully before key destruct.

V. RESEARCH METHODOLOGY

Our focus is on the key distribution algorithm, here we are using Shamir's algorithm, which is the key distribution algorithm, is the core algorithm; to make the Object storage system implementation of client distributing is done To implement a safety destruction of data with equal divided key these methods has been used. An object-based storage interface is used based on the active storage framework for storing and managing the equally divided key. System is practical to use and also preserves the privacy of data this can be seen through the functionality and security properties and the evaluation of this prototype. Reasonably low runtime overhead is imposed by the prototype system. Self destructive supports security erasing files and random encryption keys stored in a hard disk drive or solid state drive, respectively.

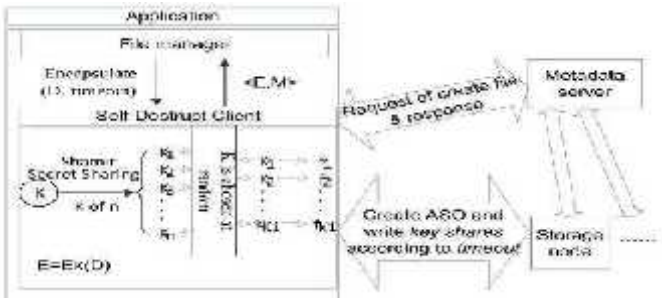

Fig 2 Key distribution using Shamir secret sharing algorithm.

VI. EVALUATION AND DISCUSSION

In this section, we are discussing the test method and implementation for SeDas and then we will give analysis on the test result. We put up a data storage file system based to implement the test for file uploading, downloading and sharing. The process to store data has no change, but encryption is needed before uploading data and the decryption is needed after downloading data. In the process of encryption and decryption, the user application program interacts with SeDas.

The evaluation platform built up supports simple file management, which includes some data process functions such as file uploading, downloading and sharing.

*1)Functional Testing:*We input the full path of file, key file, and the life time for key parts. The system encrypts data and uploads encrypted data. The life time of key parts is 150 s for a sample text file with 101 bytes. System prompts creating active object and the time required to encrypt the file , also shows that the entry is made to the database of the encrypted file and shows the TTL value we specified and the time left for the object to get destroyed and that means the up-loading file gets completed. The time to upload finally is the time to create active object. SeDas was checked and corresponded and the sample text file also was downloaded or shared successfully before key destruct.

*2)Performance Evaluation:*The difference of I/O process between SeDas and Native system is the additional encryption/decryption process which needs support from the computation resource of SeDas' client. We compare two systems: i) a self-destructing data system based on ac- tive storage framework for data privacy (SeDas for short), and ii) a conventional system without self-destructing data function (Native for short).We evaluated the latency of upload and download with two schemes (SeDas and Native) under different file sizes. Also, we evaluated the overhead of encryption and decryption with two schemes under different file sizes. We observe that SeDas increases the average latency of the Native system by 59.06% and 25.69% for the upload and download, respectively.

The reason for this performance degradation is the encryption and decryption processes introduce the overhead. The throughput decreases because upload/download processes require much more CPU computation and finishing encryption/ decryption processes in the SeDas system, compared with the Native system. SeDas reduces the throughput over the Native system by an average of 59.5% and up to

71.67% for the uploading. SeDas reduces the throughput over the Native system by an average of 30.5% and up to 50.75% for the downloading.

CONCLUSION

Data privacy has become increasingly important in the Cloud environment. The paper introduced a new concept or approach we can say for protecting data privacy, as user stored data and decryption keys may get obtained through legal or by other means by the attackers , the algorithm helps us in achieving this in our system as the decryption of data cannot take place if the keys have been destructed with which the data was encrypted .The properties of active storage framework is the novel aspect of our project.

REFERENCES

[1] L. Qin and D. Feng, "Active storage framework for object-based storage device," in Proc. IEEE 20th Int. Conf. Advanced Information Networking and Applications(AINA), 2006.

[2] Y. Zhang and D. Feng, "An active storage system for high performance computing," in Proc. 22nd Int. Conf. Advanced Information Networking and Applications (AINA), 2008, pp. 644-651.

[3] T.M.John,A.T.Ramani, and J.A.Chandy,"Active storage using object- based devices," in Proc. IEEE Int. Conf. Cluster Computing, 2008, pp. 472-478.

[4] S.W.Son, S.Lang, P.Carns, R.Ross, R.Thakur, B.Ozisikyilmaz, W.- K. Liao, and A. Choudhary, "Enabling active storage on parallel I/O software stacks,"in Proc.IEEE 26th Symp. Mass Storage Systems and Technologies (MSST),2010.

[5] Y.Xie,K.-K.Muniswamy-Reddy,D.Feng,D.D.E.Long, Y. Kang, Z. Niu, and Z. Tan, "Design and evaluation of oasis: An active storage framework based on t10 osd standard," in Proc. 27th IEEE Symp. Massive Storage Systems and Technologies (MSST), 2011.

[6] Y.Tang, P.P.C.Lee, J.C.S.Lui, and R. Perlman, "FADE: Secure overlay cloud storage with file assured deletion," in Proc. SecureComm, 2010.

[7] C.Wang, Q.Wang, K.Ren, and W. Lou, "Privacy-preserving public auditing for storage security in cloud computing," in Proc. IEEE INFOCOM, 2010.