IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 4, April 2015.
www.ijiset.com

ISSN 2348 – 7968

Self Destruction System for Protecting Data Privacy in Cloud Storage

Shankar Gadhve[1], Prof.Deveshree Naidu[2]

[1] M.TECH student, Department of Computer science and Engineering, RCOEM
Nagpur, Maharashtra, India

[2] Department of Computer science and Engineering, RCOEM
Nagpur, Maharashtra, India

Abstract

In Cloud Storage we store personal data which contain banking details such as account number, passwords, valuable notes, and other such information that can be misused by hackers. These data are copied and cached by Cloud Service Providers, often without users' authentication and control. Self-destruction system mainly aims at securing the user valuable data's privacy. All the information and their copies become destructed .In this paper, we present a system that meets this challenge through integration of active storage techniques. We implemented self destructive system through the different functionality and different security properties evaluations of this system. In addition to this the data privacy can be given to the system by encrypting the data

Keywords: *Triggering Parameter, Cloud data privacy, rules for uploading the data*

1. INTRODUCTION

As Cloud computing and mobile Internet are getting popularized, Cloud provides services which are becoming more and most important among people's life. People are requested to submit or post some personal information to the Cloud by the web. When people put their data, they subjectively hope service providers will secure policy to secure their information from leaking, so others user will not retrieve their privacy of data. As people depend more and most on the Internet and Cloud environment, security of their data and privacy is on more threaten. On the another hand, when information is being accessed, transformed and stored by the computer system or network must make cache, copy or stored it. Because these copied information are essential important for systems and the network system. As users who have no information about these copies and could not control them, so these copies can leak their data. On the another hand, their privacy data also can be leaked through Cloud service Providers, hacker' intrusion or some unauthorized actions. These problems could occurs challenges to secure people's data privacy. Personal important data stored in the Cloud may contain banking information, passwords, important notes, and other important data that could be used

2. LITERATURE REVIEW

2.1 Existing System:
A pioneering study of Vanish [2], is the system that provides the basic idea of self destructing data. In the Vanish method, a secure key is separate and stored in a point to point system with distributed hash tables. With joining and exiting of the point to point method, the system can maintain secure keys. According to parameter of point to point, the distributed hash tables will refresh every node after every eight hours. With Shamir Secret Sharing Algorithm, when we will not get limited parts of a key, it will not decrypt information or data archives with this public key, which emphasis the public key is vanished and the data cannot be recovered. 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 text messages that automatic self-destruct after a specific period of time. It includes 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. In this the key is permanently vanish, and the encrypted data key is permanently unreadable after data terminates time. In Vanish system each message is encrypted with a random key and storing share of the public key in a large, public key distributed hash tables. However, Sybil attacks may compromise the system by continuously crawling the distributed hash tables and saving each mass value before it can time out and the all around cost is some of magnitude of less than that mentioned in estimated.
Another system called FADE [4], provide contribution for the self destructing data by integrating cryptographic techniques. The data will be encrypted before sending it.

This system will delete the files and makes them unrecoverable by revoking the file access permission

2.2 Proposed System:

The self destructive system based on selfdes defines new modules, a self-destruct module that is associated with Set of rules. In this paper, self destructive system can meet with the need of self-destruct system with manageable rules while users can used this system as a general active storage system [3].

. A) Active Storage Object

An active storage system generates from a user system and has a triggering parameter value property. The time-to-live value is used to delete the self-destruct function. The time-to-live *value* [5] of a user object has the value infinite so that the user object will not be deleted until the user deletes it manually.

The triggering parameter is nothing but the time to live parameter which is used to activate the self destruction operation. The triggering parameter is decided by the user for how long the user wants the data on cloud environment and after the survival time the data which is uploaded on cloud that will be deleted automatically once the survival time will over

B) Self-Destruct Method

A self-destruct method is used to delete the data from the cloud storage as per the rules defines. User specifies the survival time and data will be deleted from the cloud environment once the survival time is over.

Figure 2 show the pseudo code for the entire process of algorithm. Pseudo code begins with the registration of the user and validates the user with user id and password. The other part of pseudo code is about the Data Process.

C) Data Process

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

i) Uploading

When user uploads a file. The file get encrypted before uploaded on cloud. User must specify the file and triggering parameter as arguments for the uploading procedure. Once the files have uploaded on the cloud storage, the data will be on the cloud only for the time which specify in triggering parameter. Once the time will over as mentioned in triggering parameter the file will be deleted automatically from the cloud environment.

ii) Downloading s*:* Any authenticated user who has proper permission can download data stored in the data storage system. The file get decrypted before when it is downloaded from the cloud.

```
Procedure upload (email,pwd)
Email: unique email id of the user.
Password: secret password for authentication.


BEGIN

Check the user_id
If  new user then
        Register the user to the system
else
        validate the user with user id and password

check for upload or download
if upload then

        upload(file, key)
else
        if triggering parameter expired
                Delete file from cloud   environment;

        else
                Download file
        endif
endif;
END;
```

Fig.1 Pseudo code of Process

```
Procedure Self destruction method (triggering parameter, File path)
Triggering parameter: Survival time (ttl)
File path: File stored on cloud

BEGIN

//Check the ttl value exist its time limit
If  ttl  value  equal to zero

        Delete the file from the cloud

Else
        Download the file from the cloud when it is
         Required.


endif;
END;
```

Fig.2 Pseudo code for self destruction Method

D) Rules of Erasing Data from Disk

We must secure the data from hacking someone who is not authorized user. To prevent the data from hacking we defines certain rules for automatic deletion of data.

Rules are as follows

i) .Data must be deleted automatically if it is older then one week.

ii) Monitor the file before uploading on the cloud to check for authorized file if it is not intended file then system will give the invalid message on the screen

Fig.2 shows the pseudo code for self destruct method. The purpose of this method is to delete the data from the cloud environment automatically.[6]

The deletion of file is depending upon the triggering parameter; triggering parameter is the time to live property. Self destruction method is the function which receives the two parameter first is file name and another is triggering parameter. File name is the name of file which is to be stored on cloud and triggering parameter is the time to live parameter.

Similarly Fig.1 shows the pseudo code for entire process which includes the registration of new user and validation of existing user. Once the valid user enter into the system. User can use the functions of the system.

3. OBJECTIVES

Objectives of Proposed System to implement a self destructing system and data privacy are as follows:-

The self destructive system defines some modules, a self-destruct method and triggering parameter. In this case, System could meet the requirements of self-destructing system with time to live property people can use this secure system as a general active system.

Our objectives are summarized as follows.

1) We emphasis on the AES core algorithm, which is used as the main algorithm to implement client (users). We use these methods to implement a safety destruct with set of rules.

2) Based on active storage framework, we use an object-based storage interface to store and manage the equally divided key.

3) Through functionality and security properties evaluation of this prototype, the results demonstrate that System is practical to use and meets all the privacy goals.

4) System supports security files and random AES encryption keys stored in a cloud storage or solid state drive, respectively.

5) Through functionality and security properties evaluation of this method, the results demonstrate that System is more reliable to use and accept all the secure goals. The method of the system can imposes reasonable low runtime overhead.

6) System supports security deleting files from cloud within a specified Rule

7) Advance encryption standard provide the data privacy

8) Set of rules provides the information to be uploaded on cloud.

4. PROPOSED PLAN OF WORK

1**.** Study of the Existing System as well as Proposed System:-

In this module we will do study of the existing system and also of the proposed system and whatever disadvantage that are in the existing system we have to remove it and have to see that it does not occur in the proposed system.

2. Development of Active Storage Framework:-

An active storage object that is derived from a user object and has the some set of rules for uploading the data on cloud. The set of rules are used to trigger the self-destruct operation. The values 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.

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 method: uploading and downloading.

i) Uploading file process: When a user uploads a file to a cloud storage and stores his key in this System, he should specify the file, the key as arguments for the uploading procedure. We assume data and key has been read from the file. The ENCRYPT procedure uses a AES encrypt algorithm or user-defined encrypt algorithm. After uploading data to storage server, key generated by AES algorithm will be used to create cloud storage object in storage node in the self destructive system.

ii) Downloading process: Any user who has intended permission can download data stored in the cloud storage system. The data must be decrypted before it is downloaded. The whole logic is implemented in code of user's application.

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

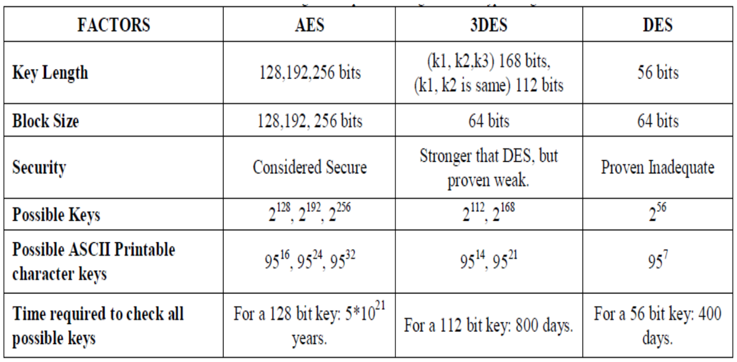
A self-destruction method is a active service method. It needs three arguments. The lun argument specifies the device; the pid argument specifies the partition and the obj_id argument specifies the object to be destructed.

5.RESEARCH METHODOLOGY

In this paper we are working on the method where there is AES encryption Algorithm is used for encryption and decryption of file and store the encrypted file on cloud in addition we also simply work on the self destruction method where we need to put the code of self destruction method. User will upload the file on cloud with any key and specify only the set of rules and data will be safe on cloud until the rules over the limit. The simple architecture for self destruction method is mentioned in fig.3
 Self destruction supports security erasing files and AES encryption keys stored in a Cloud storage or solid state drive , respectively.

Fig 3.Secure architecture of selfdes

The basic encryption and decryption of a string using AES-256 algorithm. AES, known as Rijndael, is an advanced encryption algorithm. Due to the complexity and security provided it is widely used.
Choosing a private key:
A private key is a random generated string kept on the server side. This means that only the ones who have access to the source files will know the private key. Private key is static and it won't change it's value
Generating the public key:
A public key is a random generated string that is sent to the browsers client. Because we're using AES-256 encryption we'll need a 256 bit/32 byte key to use as a cipher key. Because we're using a public encryption key we can generate it every time we want to encrypt a string:



FACTORS	AES	3DES	DES
Key Length	128,192,256 bits	(k1, k2,k3) 168 bits, (k1, k2 is same) 112 bits	56 bits
Block Size	128,192, 256 bits	64 bits	64 bits
Security	Considered Secure	Stronger that DES, but proven weak.	Proven Inadequate
Possible Keys	2^{128}, 2^{192}, 2^{256}	2^{112}, 2^{168}	2^{56}
Possible ASCII Printable character keys	95^{16}, 95^{24}, 95^{32}	95^{14}, 95^{21}	95^{7}
Time required to check all possible keys	For a 128 bit key: $5*10^{21}$ years.	For a 112 bit key: 800 days.	For a 56 bit key: 400 days.

Fig 4 Comparison with other algorithm

6 CONCLUSION

Data privacy has become increasingly important in the Cloud environment. This paper introduced a new approach for protecting data privacy from attackers who retroactively obtain, through legal or other means, a user's stored data and private decryption keys. A novel aspect of our approach is the leveraging of the essential properties of active storage framework.

REFERENCES

[1] IEEE paper on " A Self-Destructing system Based on Active Storage Framework" by: Lingfang Zeng, Shibin Chen, Qingsong Wei , and Dan Feng IEEE TRANSACTIONS ON MAGNETICS, VOL. 49, NO. 6, JUNE 2013.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] A. Devulapalli, I. T. Murugandi, D. Xu, and P. Wyckoff, 2009, Design of an intelligent object-based storage device [Online].Available:http://www.osc.edu/research/network_ file/projects/object/papers/istor-tr.pdf

[8] 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.