ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 3, September 2016

Self destruction model for protecting data in cloud storage based on data storage center

Shankar Gadhve, Prof. Deveshree Naidu
Department of CSE, R.C.O.E.M Nagpur, India

Abstract: ***In Cloud Storage we store personal data which contain banking details such as account numbers, 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 user's authentication and control. Self destruction system basic aims at securing the user valuable data on cloud. In addition we implemented self destructing system through the different functionality and different security properties. We present a system that meets this challenge through a novel integration of cryptographic techniques with KGC (Key Generation Center) and DSC (Data Storage Center). A novel approach provides additional security to data files with attributes value which is associated with each file and it requires before downloading without this value, user will not be able to download the file. The functioning of attribute value with each file is managed by data storage center. Through functionality and security properties evaluations of this system the result demonstrate that the system is practical to use and meets all the privacy goals described. Compared to the system with SEDAS, throughput for uploading and downloading with the proposed system slightly decreases by less than 65%. While latency for upload and download operations with self destruction data mechanism increases by 60%.***

Keywords: **Cloud Computing, Key Generation center, Data Privacy and security, Self destruction method.**

I. INTRODUCTION

As Cloud computing and mobile Internet is 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 [1]. When people put their data, they subjectively hope service providers will secure policy to secure their information from leaking, [2] so others user will not retrieve their privacy of data. As people depend more 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 other hand, their data also can be leaked through Cloud service Providers, hacker' intrusion or some unauthorized actions [3]. These problems could occur. Challenge is 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 and improper by an unauthorized person, a competitor, or different user. These information or data are cache, copy, and archived by Cloud Service Providers, without users' permission. Self-destructing system generally aims at securing the user data. [1] All the information and their copy become vanished or unreadable after a user-triggering time, without any user disturbance.

In this paper we are implementing a self destructing data system, which is based on data storage center. The proposed system defines three new modules. Self destruction method that is associated with each attribute key and survival parameter, Key generation center and data storage center. The self destruction system can meet the requirement of self destructing data with controllable survival time.

Our contributions are summarized as follows.

1) We focuses on key generation center which is independent system on cloud and used to generate unique key for each file .we used this system to generate key for encryption process.

.2) Based on data storage center .we use attribute value to download the data from cloud storage Attribute values are manage by data storage center.

3) Self destruction method [1] is used to delete the copies of key which is generated from KGC and DSC. Through functionality and security properties evaluation of this method, the results demonstrate that System is more reliable to use and accept all the security goals. The rest part of this paper is organized as follows. We review the literature review in Section II. We describe the design and implementation part in section III. The result cases in evaluated in Section IV , and we gives the research methodology in section V and we conclude the paper in Section VI.

II. LITERATURE REVIEW

A. Data Self Destruct

The self destructing data system in the Cloud environment should meet the following requirement.

i) How to destruct all copies of the data simultaneously and make them unreadable in case the data is out of control? A local data destruction approach will not work in the Cloud storage because the number of backups or archives of the data that is stored in the Cloud is unknown, and some nodes preserving the backup data have been offline. The clear data should become permanently unread- able because of the loss of encryption key, even if an attacker can retroactively obtain a pristine copy of that data; **ii)** No

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 3, September 2016

explicit delete actions by the user, or any third-party storing that data; **iii**) No need to modify any of the stored or archived copies of that data; **iv**) No use of secure hardware but support to completely erase data in HDD and SSD, respectively.

Tang *et al.* [11] proposed FADE which is built upon standard cryptographic techniques and assuredly deletes files to make them unrecoverable to anyone upon revocations of file access policies. Wang *et al.* [12] utilized the public key based homomorphism authenticator with random mask technique to achieve a privacy-preserving public auditing system for Cloud data storage security and uses the technique of a bilinear aggregate signature to support handling of multiple auditing tasks. Perlman *et al.* [13] present three types of assured delete: expiration time known at file creation, on-demand deletion of individual files, and custom keys for classes of data.

Vanish [1] is a system for creating messages that automatically self-destruct after a period of time. It integrates crypto- graphic techniques with global-scale, P2P; distributed hash tables (DHTs): DHTs discard data older than a certain age. The key is permanently lost, and the encrypted data is permanently unreadable after data expiration. Vanish works by encrypting each message with a random key and storing shares of the key in a large, public DHT. However, Sybil attacks [3] may compromise the system by continuously crawling the DHT and saving each stored value before it ages out and the total cost is two orders of magnitude less than that mentioned in reference [14] estimated. They can efficiently recover keys for more than 99% of Vanish messages. Wolchok *et al.* [3] concludes that public DHTs like VuzeDHT [15] probably cannot provide strong enough security for Vanish. So, Geambasu *et al.* [14] proposes two main countermeasures. Although using both OpenDHT [16] and VuzeDHT might raise the bar for an attacker, at best it can provide the maximum security derived from either system: if both DHTs are insecure, then the hybrid will also be insecure. OpenDHT is controlled by a single maintainer, who essentially functions as a trusted third party in this arrangement. It is also susceptible to attacks on roughly 200 Planet Lab [17] nodes on which it runs, most of which are housed low-security research facilities. Vanish is an interesting approach to an important privacy problem, but, in its current form, it is insecure [3].

To address the problem of Vanish discussed above, in our previous work [4], we proposed a new scheme, called *Safe Vanish*, to prevent *hopping attack*, which is one kind of the *Sybil attacks* [18], [19], by extending the length range of the key shares to in- crease the attack cost substantially, and did some improvement on the Shamir Secret Sharing algorithm [20] implemented in the Vanish system. Also, we presented an improved approach against *sniffing attacks* by way of using the public key crypto system to prevent from sniffing operations However, the use of P2P features still is the fatal weakness both for *Vanish* and *Safe Vanish*, because there is a specific at- tack against P2P methods (e.g., hopping attacks and Sybil at- tacks [3]).

In addition, for the Vanish system, the survival time of key attainment is determined by DHT system and not controllable for the user. Based on active storage framework, this paper pro- poses a distributed object-based storage system with self destructing data function. Our system combines a proactive approach in the object storage techniques and method object, using data processing capabilities of OSD to achieve data self-destruction. User can specify the key survival time of distribution key and use the settings of expanded interface to export the life cycle of a key, allowing the user to control the subjective life-cycle of private data.

III. DESIGN AND IMPLEMENTATION

A. Self destruction System architecture

Fig. 1 shows the architecture of proposed system there are three important modules**. i) Self destruct method:** This method is use to delete the key and data from the cloud storage as per priority.

ii) Key Generation center: The KGC is independent system which is used to generate the key for each individual file. **iii) Data Storage Center:** Data storage center is responsible for user management, session management and storage management.

1. Self destruct Method

Self destruction plays important role in computer science and engineering. Self destruction method is mainly aims at protecting data privacy. In self destruction method we focus on priority of data. Priorities can be set as permanent and temporary files. This method gives the survival time or we can say triggering parameter .The survival time decide how long the key and data will be survive on cloud .The survival time is given to only temporary files only, where as permanent file don't require the TTL value. This functionality provides the reliability to the user.

2. *Key Generation Center*

Key Generation center is independent system which is used to generate private keys of users by applying KGC master secret keys to user associated set of attributes [4]. Once the key generated, it is send to self destructions system. As the key generation is totally independent from main system hence this makes system more secure from the third party threat. Fig.2 show the pseudo code for the key generation process for KGC system

3. *Data Storage Center*

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 3, September 2016

Data storage center is storage center in self destruction system which is used for session management, user management and key management. DSC key is very important for encryption and decryption process. Data storage center is also used to store the data on cloud. It is important to generate attribute value for each file which is used while downloading the files from the cloud. Session management includes the individual user management on cloud. As it also used to generate DSC key. This is used for encryption process where this encryption algorithm is used to encrypt the data and decryption algorithm [5] is used the decrypt the. Data storage manages the all individual user with respect to their session for storing the data.

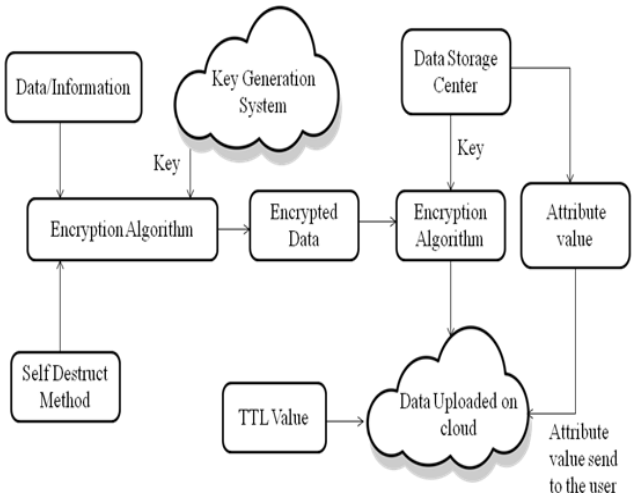


Fig.1 Self destruction system architecture

Fig.3 shows the pseudo code for DSC key generation. Key is used for encryption process, key is randomly generated.

4. Secret Attribute Value

It is secret value which is generated by data storage center [24], the value is alphanumeric this attribute value is associated with each file and it is used for download. This secret attribute key is send to the user while downloading the file [9]. Attribute value makes the system more secure.

```
Procedure KGC Key ( )
Begin
        //Variable declaration
        int code count=16
        String All Character
        String Random Code
        int temp = -1
            for I from 0 to code count then
                if (temp!=-1) then
                        Random number
                        generation
                End of if
                int t=randnext (62)
                if (temp=-1AND temp==1)
                        return secret key
                End of  if
                temp=t
                Random code =all character
                                Array
            End of loop
End of function
```

Fig.2 Pseudo code for key generation in KGC

5. Time to live value

Triggering parameter value is used to activate the self destruction system. TTL value is specified by the user.TTL value is the date and time duration for existence of temporary files on cloud.TTL value can be updated before expiration. As in implemented part user can update the TTL value before expired.

```
Procedure DSC Key ( )
Begin
        //Variable declaration
    int code count=16
    String All Character
    String Random Code
    int temp = -1
        For I from 0 to code count then
            if (temp!=-1) then
                    Random number
                    Generation
            End of if
            int t=randnext (62)
            if (temp=-1AND temp==1)
                    return secret  key
            End of  if
            temp=t
            Random code =all character
                            Array
            End of loop
End of function
```

Fig.3 Pseudo code for DSC Key

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 3, September 2016

B. Data Process

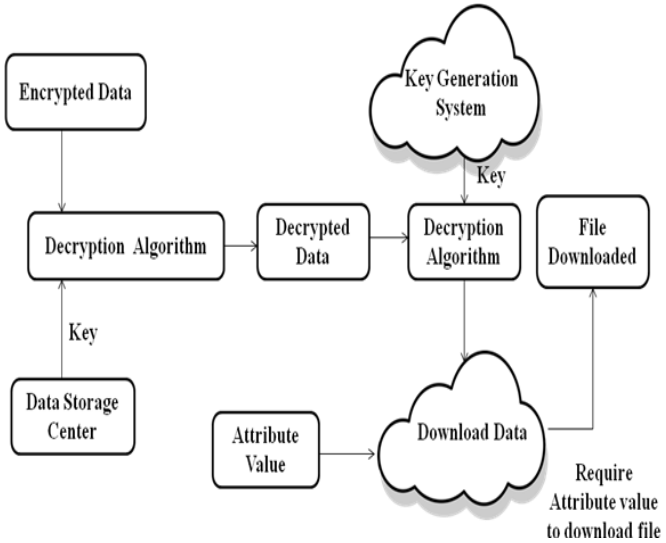
To use the self destruction system. Data storage center should implement logic of data process the different Operations are uploading and downloading. Fig.4 shows the flow of data process in data storage

1. Uploading

Uploads a file and triggering parameter on a cloud storage system. 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 storage environment only in temporary files if the priorities are permanent file then there is no need of triggering parameter. [28]

2. Downloading:

In downloading proves only authenticated user who has proper permission can download data stored in the data storage center but before downloading the file user need to enter proper authenticated key for file download [10]. Key will be receiving from registered email id. Data storage center.



.

Fig 4. Data Process (Uploading and downloading file)

C. Network Traffic Analysis Tool

It is independent tool design for self destruction system. This tool is used to measure overhead, time latency and throughput. It analyzes the data during download and uploading process. It is independent tool which get the result of latency, overhead and throughput [30]. This tool gives the result of all types of files which are downloaded on cloud and vice versa. The result is totally depend on the speed of connection.Fig.5 shows the graphical representation of data which is shown by network traffic analysis where user can see the latency and throughput and overhead of each file in cloud.

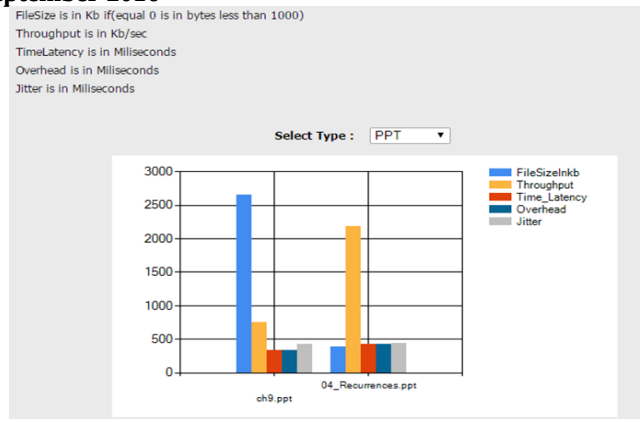


Fig. 5 Graphical representation of data

IV. EVALUATION AND TEST CASES

In this part, we discuss the test cases and implementation for system and then give the analysis on the test result. We put up the data storage center and key generation center on cloud to implement for file uploading and downloading.

A. Experimental Setup

There are multiple storage services for a user to store data. Meanwhile, to avoid the problem produced by the centralized "trusted" third party, the responsibility of KGC i s to generate the Key and send to Data storage center and DSC also generate the DSC key both provide the key and generate new key for encryption process this makes the system secure from third party [25] The result is taken out by considering different types of where store on cloud and result is taken out from network analysis tool. The result cases show the uploading and downloading process, in which the result is totally better as compare to traditional system (without Self Destruction System).

B. Performances Evolution

As mentioned in experimental setup a self destruction data system based on data storage center. We evaluated the latency of upload and download under the different file sizes. We observed that in result throughput is decreases and latency is increases fig.6 shows the comparisons and download operation [27].

V. RESEARCH METHODOLOGY

In this paper we are working on the method i.e self destruction method, key generation Center (KGC) and Data Storage Center (DSC).[26] The self destruction method id used to delete the temporary data from cloud with the help triggering parameter. The KGC is the key generation center this approach allow user to generate random key from independent system on cloud i.e. KGC [30]. This key where send to data storage to create a unique key for encryption process and Data storage center is responsible for session management, user management and key management. It is also used for data storage in cloud .We have implemented a unique approach in this

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 3, September 2016

system which make the system more efficient and secure in point of security concern. User will upload the file on cloud with triggering parameter for temporary file and store the data on cloud where as for download process user need to put the key for download the file from cloud without any key as mentioned user can not download the file from cloud this make the system more secure and this whole concept is new and one more thing which is added in this implementation part is Network analysis tool which help the user to get the correct value to evaluate the test cases for all type of file in Self destruction system. Network analysis tool gives the graphical representation of all types of data which are stored on cloud.

Fig.6 (a) Comparison of throughput in the upload

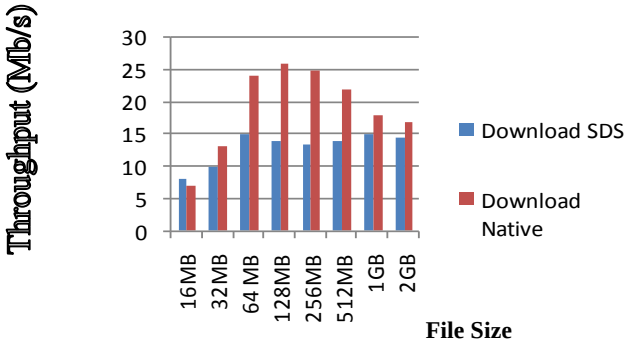


Fig.6 (b) Comparison of throughput in the Download

VI. CONCLUSION & FUTURE ENHANCEMENT

Data security has become important in the cloud storage. This paper introduced a new method of protecting data in cloud environment. This system causes important information such as Personal data, information in organization, banking sector, valuable notes and important information which can be stored on cloud. Data storage center module is used to store the data on cloud and delete the data from the cloud only in case of temporary files which gives he lower load of storage on cloud. The plan is to release the current model system will help to provide researchers with further valuable experience to inform future data storage center for Cloud services The proposed model can be extended with fixed bandwidth to increase the uploading and downloading speed of files.

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] S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Halderman, C. J. Rossbach, B. Waters, and E. Witchel, "Defeating vanish with low-cost sybil attacks against large DHEs," in Proc. Network and Distributed System Security Symp., 2010..

[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]] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, Mediated Cipher text-Policy Attribute-Based Encryption and Its Application," Proc. Int'l Workshop Information Security Applications(WISA '09), pp. 309-323, 2009.

[6] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc.Int'l Conf. Theory and Applications of Cryptographic Techniques (Eurocrypt '05), pp. 457-473, 2005.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,"Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Cipher text-Policy Attribute Based Encryption," Proc. IEEE Symp. Security and Privacy, pp. 321-334, 2007.

[9] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-Based Encryption with Non-Monotonic Access Structures," Proc. ACM Conf. Computer and Comm. Security, pp. 195-203, 2007.

[10] 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.

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

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

[13] R. Perlman, "File system design with assured delete," in Proc. Third IEEE Int. Security Storage Workshop (SISW), 2005.

[14] R. Geambasu, J. Falkner, P. Gardner, T. Kohno, A.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 6, Issue 3, September 2016

Krishnamurthy, and H. M. Levy, Experiences building security applications on DHTs UW-CSE-09-09-01, 2009, Tech. Rep.. Azureus, 2010 [Online]. Available: http://www.vuze.com/

[15] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S Shenker, I. Stoica, and H. Yu, "OpenDHT: A public DHT service and its uses," in Proc. ACM SIGCOMM, 2005.

[16] J. R. Douceur, "The sybil attack," in Proc. IPTPS '01: Revised Papers from the First Int. Workshop on Peer-to-Peer Systems, 2002.

[17] B. Poettering, 2006, SSSS: Shamir's Secret Sharing Scheme [Online]Available: http://point-at-infinity.org/ssss/

[18] A. Lewko, A. Sahai, and B. Waters, "Revocation Systems with Very Small Private Keys," Proc. IEEE Symp. Security and Privacy, pp. 273-285, 2010.

[19] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-Based Encryption with Efficient Revocation," Proc. ACM Conf. Computer and Comm. Security, pp. 417-426, 2008.

[20] N. Attrapadung and H. Imai, "Conjunctive Broadcast and Attribute-Based Encryption," Proc. Int'l Conf. Palo Alto on Pairing-Based Cryptography (Pairing), pp. 248-265, 2009.

[21] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure Attribute-Based Systems," Proc. ACM Conf. Computer and Comm. Security, 2006.

[22] S. Rafaeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication," ACM Computing Surveys, vol. 35, no. 3, pp. 309-329, 2003.

[23] P. Golle, J. Staddon, M. Gagne, and P. Rasmussen, "A Content- Driven Access Control System," Proc. Symp. Identity and Trust on the Internet, pp. 26-35, 2008.

[24] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS '10), 2010.

[25] S.D.C. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-Encryption: Management of Access Control Evolution on Outsourced Data," Proc. Int'l Conf. Very arge Data Bases (VLDB '07), 2007.

[26] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 213-229, 2001.

[27] A. Kate, G. Zaverucha, and I. Goldberg, "Pairing-Based Onion Routing," Proc. Privacy Enhancing Technologies Symp, pp. 95-112, 2007.