ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

A Survey on Self-Destruction Data System Based on Active Storage Framework

Sonali P. Gathe, Prof. Manoj S. Chaudhari

M.TECH Student, Department of CSE, Priyadarshini Bhagwati College of Engg., Nagpur, India

Prof. & Head of the Department, Department of CSE, Priyadarshini Bhagwati College of Engg., Nagpur, India

ABSTRACT: As Cloud computing and mobile Internet is getting popularized, Cloud services are becoming more and more important in people's life. People are requested to submit or post some personal private information to the Cloud by the Internet. When people post their data, they subjectively hope service providers will provide security policy to protect their data from leaking, so others people will not invade their privacy, users generally stores their personal information on cloud. Later that information can be cached, copy, archived by the CSP's (Cloud Service Provider). Self-destructing data is considering as spy proof future of the internet, by using this data is like a time bomb on the network, it means that after certain time interval. Primarily self-destruction of data was introduced on DHT (Distributed Hash Table), since low cost Sybil attack on DHT are the problems. Self-destructing data is now on active storage framework (SeDas) because it giving higher security and performance. Also uploading and downloading of the file having very poor performance especially with the large files. A time-to-live (TTL) is integrated into the executable to provide an additional layer of security so that the data is only accessible within a defined time period.

KEYWORDS: Self Destruction of Data, Cryptography, Active Storage

I. INTRODUCTION

Now a days more and more services and applications are emerging in the Internet, exposing sensitive electronic data in the internet has become easier. With development of Cloud computing and popularization of mobile Internet, Cloud services are becoming more and more important for people's life. People are more or less requested to submit or post some personal private information to the Cloud by the Internet. As people rely more and more on the Internet and Cloud technology, security of their privacy takes more and more risks. On the one hand, when data is being processed, transformed and stored by the current computer system or network, systems or network must cache, copy or archive it. These copies are essential for systems and the network. As people have no knowledge about these copies and cannot control them, so these copies may leak their privacy. On the other hand, their privacy also can be leaked via Cloud Service Providers negligence, hackers' intrusion or some legal actions. These problems present formidable challenges to protect people's privacy. Personal data stored in the Cloud may contain account numbers, passwords, notes, and other important information that could be used and misused by a miscreant, a competitor, or a court of law. These data are cached, copied, and archived by Cloud Service Providers, often without users' authorization and control. Self-destructing data mainly aims at protecting the user data's privacy. All the data and their copies become destructed or unreadable after a user- specified time, without any user intervention. In addition, the decryption key is destructed after the user-specified time. Self destruction is implemented by encrypting data with a key and then retrieving the information needed to reconstruct the decryption key with one or more third parties[12]. Assuming that the key reconstruction information disappears from the retrieval with trust from third parties at the intended time, encrypted data will become permanently unreadable.

(1) even if an attacker retrieve a copy of the encrypted data and the user's cryptographic keys and passphrases after the timeout.
(2) Without the user or user's agent taking any precise action [5] to delete it.
(3) with no need to alter any stored or archived copies of that data.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

A. Self-Destructing Data System :

Control over data lifetime will become more important as more public and private activities are captured in digital form, whether in the cloud or on personal devices[11]. Self destructing data systems can help users some control, by ensuring that data becomes permanently unavailable after certain period of time.

As people getting more dependent on the Internet and Cloud technology, safety of their privacy takes at very high risks[10]. First when data is being transformed, processed and stored by the current node or network node must cache, copy or archive it. These copies are vital for systems and the network. However, people are unknown about these copies and cannot control them, so these copies may getaway their privacy[12]. And second their privacy also can be leaked via Cloud Service Providers (CSPs') carelessness, hackers' intrusion or some legal actions.

These problems present dreadful challenges to protect people's privacy [1]. The self-destructing data system in the Cloud environment should meet the following requirements: i) How to destruct all copies of the data. Ii) No explicit delete actions by the user, or any third-party storing that data. Iii) No compelling reason to adjust any of the saved or documented copies of that data. Iv) No use of secure hardware but support to completely erase data in HDD and SSD, respectively.

B. Self Destruction using Time to Live property:

Time to Live is a mechanism that limit the lifespan or lifetime of keys stored in cloud. TTL may be as a counter or timestamp attached to or embedded in the keys. Once the prescribed event occur or timespan has elapsed, keys can be self destructed without any user intervention. In computing application, TTL is used to improve performance of caching or to improve privacy from the leakages.

II. BACKGROUND

Computing capabilities can be delivered as a service by using some phenomenons of distributed, grid and elastic nature introduced in cloud technology. This computing makes the load reduction along with improve computing experiences on browser rather than some stand alone host machines. It is made feasible by using virtualizations and resource pooling. Data is the most valuable asset and the rest of the things are supportive actors for making the data transitions or exchanges, more efficient and secure. All it needs data to be processed by various nodes, has to traverse through various networks, stored on different devices, work simultaneously on multiple copies of data.

After the usage period of data is over along with its lifecycle, it should be removed from each and every entity. Normally the lifecycle management includes the production, transfer, use, share, archive and deleted. The information which is mainly used and public will stays for longer phase and the data with fewer use will detached more recurrently. But in contemporary scenarios there are no such guidelines obtainable for effective data demolition. It could be named in several ways by different authors like destruction, deletion, removal, decommissioning, sanitizing, vanishing, disposal etc. Removing the complete copy of data is a destruction activity and is based on the futuristic aspects and time which defines the scope of its usability.

Also during the fault tolerant procedure the system normally replicates the copy of data to several locations and after the recovery these temporary or permanent copies needs to be removed. At the time point of deletion points, some files and their metadata residues are remains at the different locations which later be used for some attack initiations or might compromises the security of the system. Such issues are not taken over in the current data destruction in lifecycle management or storage schemes. Even though, the cloud computing and other web based computing are adapted very rapidly, there are some factors which are crossing the juridical limits of information processing. Mainly they are affecting the confidentiality and integrity of data.

These pre-empt data regeneration from its residues from some forensic means likewise given in act of enforcement in [3]. Some more guidelines are available with ISO standards like 270001 and NIST standards of demolishing data

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

completely after usability period is over [4]. Some solution of the above mentioned problem has to be provided for improved security and the trust over the system. For making the improvements in the security of remaining copies of the data some policies related to the data removal and its metadata structure usages needs to be defined [5].

III. LITERATURE SURVEY

1. Lingfang Zeng , Shibin Chen , Qingsong Wei , and Dan Feng SEDAS: "A self-destructing data system based on active storage framework" - This paper proposes a distributed object-based storage system with self-destructing knowledge operate. we tend to use SeDas system with the assistance of Shamir's algorithmic program for secure fund dealings. methodology} combines a proactive approach within the object storage techniques and method object, victimization processing capabilities of OSD to attain knowledge self-destruction. User will specify the key survival time of distribution key and use the settings of swollen interface to export the life cycle of a key, permitting the user to manage the subjective life-cycle of personal knowledge. Vanish may be a system for making messages that mechanically destroy once a amount of your time. The secret is for good lost, and also the encrypted knowledge is for good unclear once knowledge expiration.

Vanish is a motivating approach to a very important privacy downside, but, in its current kind, it's insecure. Vanish is that the previous approach of the Sedas System, it additionally supported key generation algorithmic program however at a time it generate just one key thus rather than that Sedas generate multiple keys with the assistance of shamirs algorithmic program thus its higher for security purpose. Also, we tend to bestowed Associate in Nursing improved approach against sniffing attacks by means of victimization the general public key cryptosystem to stop from sniffing operations.

2. Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "FADE: Secure overlay cloud storage with file assured deletion," FADE, was proposed by tang et al provides a 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 permissions. Another system called File System Design with assured delete proposes three types of file delete. First is the expiration known at the time of file creation, second is on demand deletion of individual files and third is the usage of custom keys for classes of data. As given above, many systems have been proposed to implement a self destructing system among which only some provide promising results. The system doesn't have a user controllable data expiration time. They rather have a fixed time for file expiration which is not an efficient approach for the self destructing scenario.

3. R. Geambasu, T. Kohno, A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," Vanish is the system that provides the basic idea of self destructing data. The system developed is a prototype which is implemented using Distributed Hash Table (DHT). It used bittorrents Vuze DHT that can support eight hours timeout or Planet Lab hosted Open DHT that can support one week timeout. This system provides a plug-in for Firefox browser that creates a message which automatically disappears after a specified period of time. Here the expiry time for the data is controlled by the DHT and not by the user. Later many extensions are been implemented on the Vanish system

4. J.A.Chandy, M.John and T.Ramani "An Active Storage System for High Performance Computing", Traditional active memory device execute custom application code on large amount of knowledge by utilizing the unused process power of the storage nodes for computation intensive application, the performance could be quite low thanks to insufficient process power of storage nodes.

5. H.Chai, D.Feng, C.Li and K.Zhou "Implementing and Evaluating Security Control for Object Based Storage System" The development of high performance computing has based on storage capacity and I/O performance, storage system has entered the peta byte era. The storage system scale of high performance computing is very large, the amount of storage nodes is very huge.

6. Carns, P.Choudhary, S.Lang, B.Ozisikyilmaz and S.W.Son "Enabling Active Storage on Parallel I/O Software Stacks" As data sizes continue to increase the concept of active storage is well fitted for many data analysis kernals.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

The sedas system propose and evaluate an active storage system that allow data analysis, mining and statistical operations to be executed from with in a parallel I/O interface.

7. Disha Handa, Bhanu Kapoor PARC4: "High Performance Implementation of RC4 Cryptographic Algorithm using Parallelism" India RC4 is ideal for storing information that is highly sensitive and highly important. A RC4 method can secure a secret over multiple servers and remain recoverable despite multiple server failure. The dealer may act as several district participants, distributing the shares among the participants. Each share may be stored on a different server, but the dealer can recover the secret even if several servers break down as long as they can recover. The algorithm divides a message into fix sized large blocks and encrypts these blocks concurrently on multi core machine.

IV. RELATED WORK

A pioneering study of Vanish supplies a new idea for sharing and protecting privacy. In the Vanish system, a secret key is divided and stored in a point to point system with distributed hash tables. With joining and exiting of the point to point node, the system can maintain secret keys. According to characteristics 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 enough parts of a key, he will not decrypt data encrypted with this key, which means the key is destroyed 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 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. In this the key is permanently lost, and the encrypted data is permanently unreadable after data expiration time. In Vanish system each message is encrypted with a random key and storing share of the key in a large, public distributed hash tables. 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. In this case, self destructive system can meet the requirements of self-destructing data with controllable survival time while users can use this system as a general object storage system.

A. Self destructive system:

There are three parties based on the active storage framework.

1) *Metadata server:* It is responsible for user management, server management, session management and file metadata management.
2) *Application node:* The application node is a client to use storage service of the self destructive.
3) *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.

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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

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.

Uploading file process: When a user uploads a file to a storage system and stores his key in this System, he has to specify the file, the key and time-to-live as arguments for the uploading procedure. 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.

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.

V. CONCLUSION

Data privacy has become extremely important in the Cloud environment. In this paper, the methods to solve the issue of self-destruction of data on networks have been summarized. The object interface offers storage that is secure and easy to share across platforms, but also high- performance, thereby eliminating the common trade-off between files and blocks. Furthermore, objects provide the storage device with an awareness of the storage application and enable more intelligence in the device. Although there are many proposed solutions so far, there is no perfect solution yet. Many proposals are still not mature and need further research and experimentation. Besides these proposals, designing novel and suitable network architecture for self-destruction and Active storage and to improve its performance is also a hot research topic. Proposed approach is using active storage framework for maximum advantage and it is expected that after successful implementation, performance should be more than existing systems. With this privacy can be achieved in a public network.

REFERENCES

1. Lingfang Zeng, Shibin Chen, Qingsong Wei, and Dan Feng, "A Self-Destructing system Based on Active Storage Framework," *IEEE Trans. Magnetics*., vol. 49, no. 6, July 2013.
2. 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 T10OSD standard," Massive Storage Systems and Technologies (MSST), *27th IEEE Symp.* , 2011.
3. Y. Kang , J. Yang and E. L. Miller "Object-based SCM: An efficient interface for storage class memories", *Proc. 27th IEEE Symp. Massive Storage Systems and Technologies (MSST)* , 2011
4. M. Wei , L. M. Grupp , F. E. Spada and S. Swanson "Reliably erasing data from flash-based solid state drives", *Proc. 9th USENIX Conf. File and Storage Technologies (FAST)* , 2011
5. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for storage security in cloud computing," *IEEE Infocom*, 2010.
6. 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," *Proc. IEEE 26th Symp. Mass Storage Systems and Technologies (MSST)*, 2010.
7. 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", *Proc. Network and Distributed System Security Symp.* , 2010
8. Y. Tang , P. P. C. Lee , J. C. S. Lui and R. Perlman "FADE: Secure overlay cloud storage with file assured deletion", *Proc. SecureComm* , 2010.
9. L. Zeng , Z. Shi , S. Xu and D. Feng "Safevanish: An improved data self-destruction for protecting data privacy", *Proc. Second Int. Conf. Cloud Computing Technology and Science (CloudCom)* , pp.521 -528 2010

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

10. R. Geambasu, T. Kohno, A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self- destructing data," *Proc. USENIX Security Symp*., Aug. 2009, pp. 299–315.
11. A. Devulapalli, I. T. Murugandi, D. Xu, and P. Wyckoff, *Design of an object-based storage device* 2009 [Online]Available: http://www.osc.edu/research /network_ file/projects/object/papers/istor-tr.pdf.
12. Y. Zhang and D. Feng, "An active storage system for high performance computing", *Proc. 22nd Int. Conf. Advanced Information Networking and Applications (AINA)*, pp. 644-651, 2008.
13. B. Welch, M. Unangst, Z. Abbasi, G. Gibson, B. Mueller, J. Small, J. Zelenka, and B. Zhou, "Scalable performance of the panasas parallel file system," in *Proc. 6th USENIX Conf. File and Storage Technologies (FAST)* , 2008.
14. T. M. John, A. T. Ramani, and J. A. Chandy, "Active storage using object-based devices," *In Proc. IEEE International Conference Cluster Computing*, 2008, pp. 472-478.
15. L. Qin and D. Feng "Active storage framework for object-based storage device", *Proc. IEEE 20th Int. Conf. Advanced Information Networking and Applications (AINA)* , 2006
16. S. A.Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," *Proc. 7th Symp Operating Systems Design and Implementation (OSDI)* , 2006.