International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Enhancing Security in Cloud by Self Destruction Mechanism

Kshama D. Bothra[1], Sudipta Giri[2]

[1]SPP University, M.I.T. College of Engineering, Paud Road, Kothrud, Maharashtra 411038, Pune, Maharashtra, India

[2]Professor, M.I.T. College of Engineering, Paud Road, Kothrud, Maharashtra 411038, Pune, Maharashtra, India

Abstract: ***Cloud computing, a recent computing technology entirely changed the IT industry. Large amount of data can be stored in cloud storage system. Security is the prime concern for this large amount of data. Without, knowledge of authorized client, data can be viewed by other user. This data contain personal information like, account number, password and notes. All the data and their copies become self-destructed after user specified time, without any user intervention. Shamir secret sharing algorithm is used, which generates a pair of keys. Self-destruction method is consociated with time to live (TTL) property to specify the life time of the keys. After user specified time (TTL) data and its keys becomes destructed or unreadable. Self-destruction mechanism helps reducing overhead during upload and download process in cloud.***

Keywords: Cloud computing, self-destruction, Active Storage Object, Time to live (ttl), data privacy

1. Introduction

Cloud computing is a way to increase the capacity or capabilities dynamically without investing in a new framework, licensing new software, or training new personnel. Cloud services are becoming more important in people's life. Advances of Cloud computing and popularization of mobile Internet attract people and they submit or post some personal private information to cloud by internet. While doing this, they hope that service provider will provide security to their data from leaking so that third party won't invade their privacy.

More reliance of people to store their data on cloud has become prime concern. Copies provided, while processing and storing data in computer system or network system, are essential for computer system and network. However, these user don't have knowledge about these copies and hence cannot control. Copies stored in cloud environment may leak their privacy. Privacy can also be leaked via Cloud Service Provider's (CSPs) negligence, hackers or through some other legal actions. Y. Tang, Aims for designing an approach to achieve flexible access control and large-scale dynamic data management in a high secure and efficient way. This system provides secure overlay cloud storage with file assured deletion. FADE is suitable option for large scale enterprise that need's to archive large files with large amount of data[3]. Perlman *et al.* present three types of assured delete: custom keys for classes of data, on-demand deletion of individual files, and. expiration time known at file creation [8].

Vanish supplies new idea for sharing and protecting data privacy. In vanish system, a secret key is divided and stored in a P2P system with distributed hash tables (DHTs) [1]. P2P node has property of refreshing every node after eight hours. When user cannot get enough parts of a key, he/she will not be able to decrypt data encrypted by the key, which means key will be destroyed. S. Wolchok, proposes a two Sybil attack against the current Vanish system, which stores its encryption keys in the million-node Vuze Bit Torrent DHT. This attack work by continuously crawling the DHT and saving each stored value before it ages out [7]. The attack can efficiently recover about 99% of Vanish messages.

Lingfang Zeng, proposed Safe-Vanish, to prevent hopping attacks by way of extending the length range of the key shares to increase the attack cost substantially [2]. Improvement was done on the Shamir Secret Sharing algorithm in the original Vanish system [5]. Improved approach against sniffing attack was presented by using public key cryptosystem.

Attacks to the characteristics of P2P are a challenge of Vanish. Duration of key survival is also one of the disadvantage of Vanish. Considering the disadvantages Lingfang Zeng, proposed SeDas. SeDas is based on active storage framework. SeDas system has two modules, self-destruct method object that is associated with each secret key part and survival time parameter for each secret key part[4]. SeDas can meet the requirement of self-destructing data with controllable survival time. Self-Destructing data should meet the following requirement -

1) SeDas system focuses on Shamir's algorithm. Shamir's algorithm is used as a core algorithm to implement client distributing keys in an object storage system.
2) SeDas a novel system meets all the privacy preserving goals.
3) SeDas should support to completely erase data in HDD and SDD

2. Literature Survey

The following section gives the related work carried out so far in the scope of self-destructing data-

A. Vanish: Increasing Data Privacy with Self-Destructing Data

Vanish, provides the basic idea of self-destructing data. The system is a prototype which is implemented using two Distributed Hash Table, Vuze DHT and OpenDHT. Vuze

DHT support eight hours timeout while Open DHT supports one week timeout. Vanish system provides a plug-in for Firefox browser that creates a message which automatically disappears after a specified period of time. The expiry time for data is controlled by DHT and not by user. In Vanish, each message is encrypted with random key and shares of the key are stored in a large, public DHT. Sybil attacks may compromise the system by continuously crawling DHT and saving each value before it ages out. It is found that 99% of Vanish messages can be recovered. Many modifications were done on Vanish system.

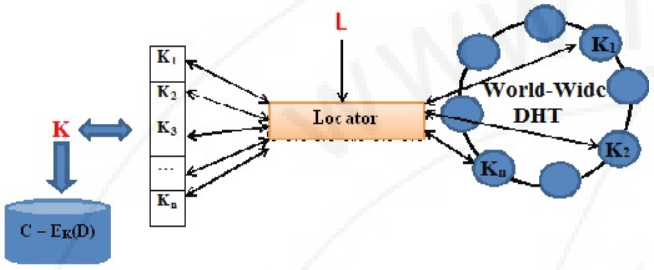

Fig 1: Vanish System Architecture

B. SafeVanish: An Improved Data Self-Destruction for Protecting Data Privacy

To address problems of Vanish a new scheme, called SafeVanish was proposed, to prevent hopping attack. Length range of key shares is extended to increase the attack cost, and improvement were done on Shamir Secret Sharing algorithm. In Vanish if attacker views user's traffic to the DHT, he can snoop and record shares while they are stored. Public key cryptography is used in safe vanish. Sender will encrypt the key shares by his private key. Then use receiver's public key to encrypt key shares.

Encrypted ciphertext key shares are then dispatch safely to the DHT node. Receiver after receiving ciphertext first decrypts it using receiver's private key, and then using sender's public key. If attacker captures ciphertext data by snooping then attacker won't be able to decrypt it since attacker don't have private key corresponding to public key. The use of P2P features still is a fatal weakness for vanish and safe vanish as there is specific attack against P2P methods.

C. SeDas: A Self-Destructing Data System Based on Active Storage Framework

Self-Destructing data mainly aims at protecting privacy by destructing data after user specified time period, without any user intervention. SeDas was used to meet this challenge through novel integration of cryptographic techniques with active storage techniques based on T10 OSD standard. There are three parties based on active storage framework. **i) Metadata server (MDS):** MDS is responsible for session management, user management, file metadata management and server management. **ii) Application node:** Application node is a client to use storage service of SeDas. **iii) Storage node:** Storage node is Object Storage system.

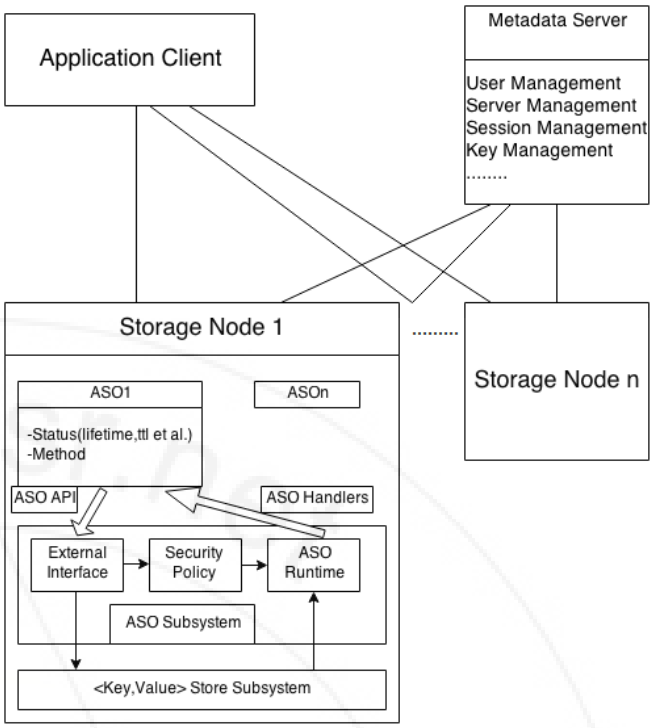

Figure 2: SeDas System Architecture

Application client can perform two operation uploading file and downloading file. When user uploads file he should specify its file, the key and ttl field. User defined encrypt algorithm can be used for ENCRYPT procedure. Shamir Secret Sharing algorithm is applied on data and key shares are generated. These key shares will be used to create active storage object in storage node in the SeDas system. User who has permission can download data stored in storage system. Data must be decrypted before use.

```
PROCEDURE UploadFile(data, key, ttl)
data: data read from this file to be uploaded
key: data read from the key
ttl: time-to-live of the key

BEGIN
   //encrypt the input data with the key
   buffer = ENCRYPT(data, key);
   connect to a data storage server;
   if failed then return fail;
   create file in the data storage server and write buffer into it;
   //use ShamirSecretSharing algorithm to get key shares
   // k is count of data servers in the SeDas System
   sharedkeys[1....k] = ShamirSecretSharingSplit(n, k, key);
   for i from 1 to k then
      connect to DS[i];
      if successful then create_object(sharedkeys[i], ttl);
      else
         for j from 1 to i then
            delete key shares created before this one;
         endfor
         return fail;
      endif
   endfor
   return successful;
END
```

Figure 3: Uploading file(pseudocode)

3. Conclusion

Due to incessant use of cloud computing for storing data, data privacy has become very important in cloud environment. Data is stored in storage devices hence much

importance is given to make storage devices intelligent and these storage devices are referred as active storage devices. In Vanish key is divided and stored in distributed hash tables. Vanish messages can be decrypted by continuously crawling DHT and saving its value. Safe-Vanish is proposed to prevent hopping attack, by extending the length range of key shares. Public key cryptography is used to prevent from sniffing attack. For this reason SeDas was introduced which is based on active storage framework. Sensitive information, such as account numbers, password self-destruct after user specified time. Result on SeDas shows that throughput for uploading and downloading decreases by less than 72%, while latency for uploading and downloading operations increases by less than 60%.

Future work will include using Hadoop to store encrypted data. It will have Metadata server, User layer, Security layer and Storage layer.

References

[1] Roxana Geambasu, Tadayoshi Kohno, Amit Levy, Henry M. Levy, "Vanish: Increasing Data Privacy with Self-Destructing Data", *In Proc. of the USENIX Security Symposium*, Montreal, Canada, pp. 299–315, August 2009.

[2] L. Zeng, Z. Shi, S. Xu, and D. Feng, "Safevanish: An improved data self-destruction for protecting data privacy," in *Proc Second Int.Conf. Cloud Computing Technology and Science(CloudCom)*, Indianapolis, IN, USA, Dec. 2010, pp.521–528.

[3] 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.

[4] L. Zeng , S. Chen , Q. Wei , and D. Feng, "SeDas: A Self-Destructing Data System Based on Active Storage Framework", in Proc. IEEE TRANSACTIONS ON MAGNETICS, VOL. 49, NO. 6, JUNE, 2013.

[5] A.Shamir, "How to share a secret," *Communication ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[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," in *Proc. IEEE 26th Symp. Mass Storage Systems and Technologies(MSST)*, 2010. 2554.

[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," in *Proc. Network and Distributed System Security Symp.*, 2010.

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

[9] 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.

[10] M. Wei, L. M. Grupp, F. E. Spada, and S. Swanson, "Reliably erasing data from flash-based solid state drives," in *Proc. 9th USENIX Conf. File and Storage Technologies (FAST)*, San Jose, CA, USA, Feb. 2011.

Author Profile

Kshama Bothra Research Scholor, MIT college of Engineering, University of Pune. He has received the B.E. degree in Information Technology from Raisoni College of Engineering, Nagpur in 2012 and currently pursuing M.E from M.I.T College of Engineering, Pune.

Prof. Sudipta Giri done MTech from IISc Banglore. He is currently working as Assistant Professor in MIT College of Engineering, Information Technology department, Pune Has received the Btech degree from IIT Kanpur.