Privacy Protection of Data in Cloud by Destroying Data at Predefined Time

Nandini S.B
Post graduation student,
E&C department, SJCE, Mysore

Shankaraiah N, PhD
ECE Department,
Research Supervisor
SJCE, Mysore.

Jayaram M.N, PhD
Professor, SJCE,
Mysore.

ABSTRACT:

Presently technical and legal landscape presents formidable challenges in data privacy. The personal data's like password, account number and any other important information's are cached, copied and achieved by the service providers. destroying of data mainly aims at providing data privacy. The data is deleted after the specified time i.e., the decryption key is deleted so that after the specified time the data cannot be accessed by any, in other words the data is deleted. Our research seeks to protect the privacy of past, archived data such as copies of emails maintained by an email provider against accidental, malicious, and legal attacks. In this paper we provide a back up copy of a data is kept in a local system when the data is destructed in the cloud. and also in our research we provide a onetime password for the first login. Specifically, we wish to ensure that all copies of certain data deleted after a user specified time, without any specific action on the part of a user, and even if an attacker obtains both a cached copy of that data and the user's cryptographic keys and password.

Keywords: data privacy, self destructing, cryptographic keys and password

1. INTRODUCTION

Now a day's people relay more on the internet. Many times people are requested to enter their personal details in the network. People are more or less requested to submit or post some personal private information to the Cloud by the Internet all these data's will be stored in the cloud. the people will be in the intension that service provider will provide the security. there may be chances of leaking of data legally or illegally.The goal of this project is to provide security to our data's which is prompted into the cloud and also facilitates the user with the backup copy of the data after destruction in cloud. In this system the data is deleted by itself after a specified time, i.e., after it is no longer used. it should be done automatically without external action i.e., without the intervention of the user or the third party who is storing the data. In this system after a specified time the key used to decrypt the data will get destructed. numerous applications are benefited from this privacy system. here in additionally we keep a back up copy of the data in our local disk after it is deleted. and we provide a password, which is generated for the initial login to the cloud.

Today's web centric world the destruction of data after it is no longer needed is broadly applicable. With destroying the data, users can regain control over the lifetimes of their Web objects, such as private messages on social networks. A system achieving these goals would be broadly applicable in the modern digital world. Our research seeks to protect the privacy of past, archived data such as copies of emails maintained by an email provider against accidental, malicious, and legal attacks and the data is kept backup in the local system of the user, so that the user can retrieve the data after destructing it in the cloud.

In this case, by deleting data can meet the requirements of data with controllable survival time while users can use this system as a general object storage system. here we mainly focus on the key distribution algorithm, that is Shamir's secret sharing algorithm. which is used as the core algorithm to implement client (users) distributing keys in the object storage system. which intern uses the AES algorithm. We use these methods to implement a safety destruct with equal divided key .Through functionality and security properties evaluation of this technique, the results demonstrate that, it is practical to use and meets all the privacy-preserving goals.In the section 2 we deal with the survey of the previous works to provide the security. in the section 3 we deal with the existing system or what is the main aim of this paper and how it is implemented.

2. RELATED WORKS

Tang et al. 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[7]. Wang et al. 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. 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. the architecture is shown in Fig 1:

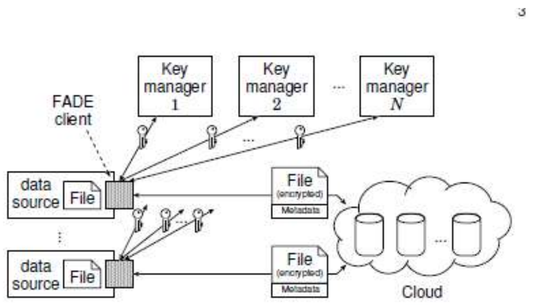

Fig 1 The fade system

but, It doesn't support operations for a batch of files. and Along with the file, it has to transmit metadata.

2.1 Vanish System

Vanish is a system for creating messages that automatically self destruct after a period of time. it integrates cryptographic techniques with global-scale, P2P, distributed hash tables(DHTs)[1]. The design of DHTs varies, but DHTs like Vuze generally exhibit a put/get interface for reading and storing data, which is implemented internally by three operations: lookup, get, and store. To store data, a client first performs a lookup to determine the nodes responsible for the index it then issues a store to the responsible node, who saves that (index, value) pair in its local DHT database. DHTs discard data older than 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 string shares of the key in a large, public DHT. however, Sybil attacks may compromise the system by continuously crawling the DHT and saving each stored value before it ages out. they can effectively recover keys for more than 99% of vanish messages. the public DHTs like vuzeDHT probably cannot provide strong enough security for vanish. although using both OpenDHT and vuzeDHT might raise the bar for an attacker, at best it 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. vanish is an interesting approach to an important privacy problem, but in its current form, it is insecure. the vanish system is shown in the Fig 2:

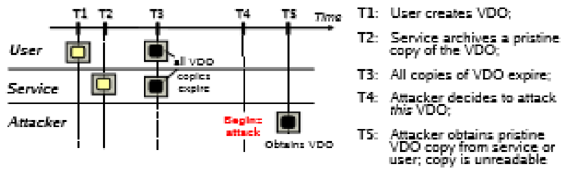
T1
T2
T3
T4
T5
Time
User
Service
Attacker
all VDO copies expire
Begins attack
Obtains VDO
T1: User creates VDO;
T2: Service archives a pristine copy of the VDO;
T3: All copies of VDO expire;
T4: Attacker decides to attack this VDO;
T5: Attacker obtains pristine VDO copy from service or user; copy is unreadable

Fig 2 The Vanish System

according to the property of DHT in some the timeout varies. if the application needs few more extra time it cannot be used for example Vuze has a fixed 8-hour timeout, while OpenDHT allows clients to choose a per-data item timeout of up to one week. Some special attacks to characteristics of P2P are challenges of Vanish, uncontrolled in how long the key can survive is also one of the disadvantages of vanish.The Sybil attacks, by extending the length range of the key shares to increase the attack cost substantially, and did some improvement in Shamir secret sharing algorithm implemented in the vanish system. also presented an improved approach against sniffing attacks by way of using the public key cryptosystem to prevent from sniffing operation. however, the use of P2P features still is the fatal weakness both for vanish and safe vanish, because there is a specific attack against P2P(peer to peer) methods(hopping attack, Sybil attack).

2.2 Safe Vanish

To address the problem of vanish. new scheme has been proposed called safe vanish[3]. to prevent hopping attack, which is one kind of the Sybil attacks(The Sybil attack in computer security is an attack wherein a reputation system is subverted by forging identities in peer-to-peer networks.), by extending the length range of the key shares to increase the attack cost substantially, and did some improvement in Shamir secret sharing algorithm implemented in the vanish system. also presented an improved approach against sniffing attacks by way of using the public key cryptosystem to prevent from sniffing operation. however, the use of P2P features still is the fatal weakness both for vanish and safe vanish, because there is a specific attack against P2P methods(hopping attack, Sybil attack).

2.3 Self destructing data system

Ling fang Zeng , Shibin Chen , Qingsong Wei , and Dan Feng, In this system the data is deleted at the predefined time. in this system the data is deleted without the intervention of the user or the third person, the data is destructed by itself. In this paper after the destruction of data we cannot get the data again and we cannot get the copy. and if in case the other changes any of the data or if there is any uploading or downloading the process carried will not aware to the owner of the data.In our research we extend the previous work by providing password for the initial login. and another addition in our research is that the data which is deleted in the cloud will be kept backup at the local system, so that the user can get the data in local system which is deleted in the cloud. and also it informs the user by sending a mail for each action carried. and in the algorithm part the additional AES is used.

3. PROPOSED SYSTEM

The privacy methods described here can meet the requirements of destroying data with controllable survival time. Security algorithms are used to encrypt the user data each time user uploads them, it secures the data. Since the data is destroyed once the “time to leave” expires the data is more secured. If the nodes where the data stored is not reachable, the data cannot be erased, but the algorithm encrypts the user data with the shared key so that data can be downloaded which is not readable. the system architecture is in Fig 3 as follows:

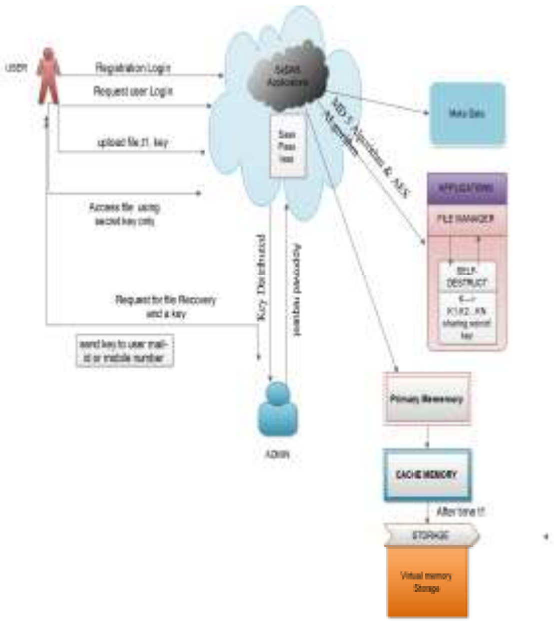

Fig **3 The system architecture**

Working: Any user can register to the platform with name, email-id. then the password is send to the specified mail id. Registered user can login to the platform with user name and password. during the initial login the user can change his password as they desire. Valid user is allowed to upload/download file to/from the cloud. User has to specify the time to live value, after which the file should be unreadable. The file and the key are stored in cache memory. After time to live value these data are removed from the cache. Then the data will be stored in virtual memory in encrypted form. Later the user can recover old file by contacting admin.

The users local system contains the backup copy of the data which he has uploaded in the cloud. and each action which is performed will be sent to the specified mail address so that the user can know about his logins and actions in his platform.
The main algorithm used here is the Shamir secret sharing algorithm. Shamir's Secret Sharing is an algorithm in cryptography created by Adi Shamir[2]. it is a form of secret sharing, where a secret is divided into parts, giving each participant its own unique part, where some of the parts or all of them are needed in order to reconstruct the secret.Counting on all participants to combine together the secret might be impractical, and therefore sometimes the threshold scheme is used where any k of the parts are sufficient to reconstruct the original secret. The goal is to divide data D (e.g., a safe combination) into n pieces $D_1,\ldots D_n$. in such a way that:

1. knowledge of any k or more D_i pieces makes D easily computable.

2.knowledge of any k-1 or fewer D_i pieces leaves D completely undetermined(in the sense that all its possible values are equally likely). this scheme is called(k,n) threshold scheme. if k=n then all participants are required to reconstruct the secret. AES(Advanced Encryption Standard) Algorithm is used for User data's Security in the cloud. AES is a symmetric block cipher with a block size of 128 bits. Key lengths can be 128 bits, 192 bits, or 256 bits; they are called AES-128, AES-192, and AES-256, respectively. AES- 128 uses 10 rounds, AES-192 uses 12 rounds, and AES-256 uses 14 rounds.

4. CONCLUSION

Data privacy has become increasingly important in the Cloud environment. The 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. it causes sensitive information, such as account numbers, passwords irreversibly deletes without any action on the user's part.The measurement and experimental security analysis sheds insight into the practicability of our approach. The plan to release the current privacy system will help to provide researchers with further valuable experience to inform future object based storage system designs for Cloud services and provide advance security to the data.

5. REFERENCES

[1] 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.

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

[3] L. Zeng, Z. Shi, S. Xu, and D. Feng, "Safe vanish: 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.

[4] 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.

[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] 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.

[8] [Online]. Available: http://www.planet-lab.org/ 9.B.Poettering,2006,SSSS:Shamir'sSecretSharingScheme[Online].

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

[10] 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.

[11] Available: http://point-at-infinity.org/ssss/R. Perlman, "File system design with assured delete," in Proc. Third IEEE Int. Security Storage Workshop (SISW), 2005.

[12] Azureus, 2010 [Online]. Available: http://www.vuze.com KRUMM, J. Inference attacks on location tracks. In Pervasive(2007).15https://en.wikipedia.org/wiki/Shamir%27s_Secret_Sharing

[13] Bowers, K.D., Juels, A., and Oprea, A.: Proofs of Retrievability: Theory and Implementation. In: Cryptology ePrint Archive, Report 2008/175 (2008), http://eprint.iacr.org/

[14] Dekel Tanke, Cloud Foundary, cited 201l: Analysis of April 25 and 26, 2011 Downtime. [Available online at http://support.cloudfoundry.com/entries/20067876-analysis-of-april-25-and-262011-downtime]

[15] Z. Yang, S. Zhong, and R. Wright, "Privacy-preserving queries on encrypted data," in Proc. of the 11 European Symposium on Research In Computer Security, 2006