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.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Time Constrained Datadestruction Using Blowfish Algorithm In Cloud

R.K.Vinothraja[1], T.Krishnakaarthik[2]

P.G Scholar, Department of information technology and Engineering, Nandha College of Technology, Erode, Tamilnadu[1]

Associate Professor, Department of information technology and Engineering, Nandha College of Technology, Erode, Tamilnadu[2]

Abstract: Account number, password and some of the important details of the client is stored in the cloud. The security for the content stored in the cloud is very important. For that the time constrained is the system used in this to give the security in the cloud.Sedas means self-destruction. The data stored in the cloud is destructed automatically when the time limit exceeded. It is used in the time of uploading and downloading the data in the cloud. Blowfish algorithm is used for the data security in the cloud storage.MAC algorithm is used for the integrity verification. Acceptably decreases by less than 72%, while latency for upload/download operations with self-destructing data mechanism increases by less than 60%.

Index Terms: Cloud computing, active storage device, data privacy

I. INTRODUCTION

Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams. Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other Cloud computing provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. 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 P2P system with distributed hash tables (DHTs). With joining and exiting of the P2P node, the system can maintain secret keys. According to characteristics of P2P, after about eight hours the DHT will refresh every node. With Shamir Secret Sharing Algorithm, when one cannot get enough parts of a key, he will not decrypt data encrypted with this key, which means the key is destroyed. Some special attacks to characteristics of P2P are challenges of Vanish, uncontrolled in how long the key can sur vive is also one of the disadvantages for Vanish.

II. DATA SELF-DESTRUCT

The self-destructing data system in the Cloud environment should meet the following requirements:

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 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 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. 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 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 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 estimated. They can efficiently recover keys for more than 99% of Vanish messages. Wolchok et al. Concludes that public DHTs like VuzeDHT probably cannot provide strong enough security for Vanish. So, Geambasu et al. Proposes two main countermeasures.

A. Object-Based Storage and Active Storage

Object-based storage (OBS) uses an object-based storage device (OSD) as the underlying storage device. The T10 OSD standard is being developed by the Storage Networking Industry Association (SNIA) and the INCITS T10 Technical Committee. Each OSD consists of a CPU, network interface, ROM, RAM, and storage device (disk or RAID subsystem) and exports a high-level data object abstraction on the top of device block read/write interface.

With the emergence of object-based interface, storage devices can take advantage of the expressive interface to achieve some cooperation between application servers and storage devices. A storage object can be a file consisting of a set of ordered log-ical data blocks, or a database containing many fi les, or just a single application record such as a database record of one trans-action. Information about data is also stored as objects, which can include the requirements of Quality of Service (QoS) [23], security [24], caching, and backup.

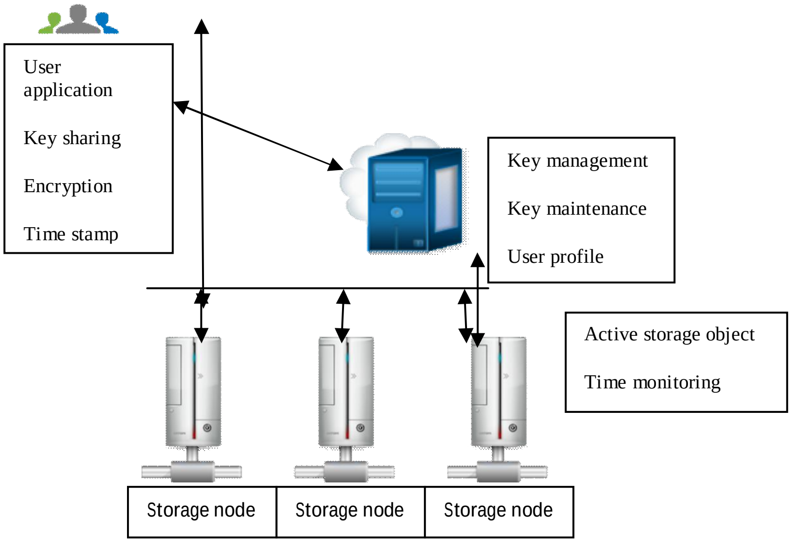

Fig 1. Architecture diagram

Important research branches in the domain of intelligent storage systems. For instance, Wickremesinghe et al.

proposed a model of load-managed active storage, which strives to integrate computation with storage access in a way that the system can predict the effects of offloading computation to Active Storage Units (ASU). Hence, applications can be configured to match hardware capabilities and load conditions. MVSS, a storage system for active storage devices, provided a single framework
To support various services at the device level. MVSS separated the deployment of services from file systems and thus allowed services to be migrated to storage devices.

B. Completely Erase Bits of Encryption Key

In erasing files, which include bits of the encryption key, is not enough when we erase/ delete a file from their storage media; it is not really gone until the areas of the disk it used are overwritten by new information. With flash-based solid state drives (SSDs), the erased file situation is even more complex due to SSDs having a very different internal architecture. Several techniques that reliably delete data from hard disks are available as built- in ATA or SCSI commands, software tools (such as, Data Wipe HDDeraseSDelete),and government standards

III. PROPOSED SYSTEM

Blowfish is used as the core algorithm to implement client distributing keys in the object storage system. Storage interface is used to store and manage the equally divided key. It supports security erasing files and random encryption keys stored in a hard disk. All the data and their copies become destructed or unreadable after a user-specified time, without any user intervention.

A. Advantage

Time constrained can meet the requirements of self-destructing data with controllable survival time while users can use this system as a general object storage system. It is practical to use and meets all the privacy-preserving goals and imposes low overhead.

IV. SYSTEM MODEL

A. User and Storage node Key Sharing

Create the user id for store the data in cloud. Generate random keys from user. For that BLOWFISH algorithms are used. Using this algorithm the user generates the keys. Sharing of keys between the Meta server and the User application. Enable N storage node for storing the data. And the user sends the request to the server. Blowfish has a 64-bit block size and a variable key length from 32 bits up to 448 bits and making it ideal for securing data. It is a variable-length key block cipher. It is suitable for applications where the key does not change often, like a communications link or an automatic file encryption. Blowfish is a symmetric block cipher that can be used as a drop-in replacement for DES or IDEA. It has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm much faster than DES and IDEA.

B. Data Transformation process

Data processing in the time constrained will have two main phases. Upload and Download. File uploading process will send the Upload request to the server. Then the server gives permission to upload the data to the client. After accepting get the N keys randomly from the user key. Then the client split the data according to the size of the storage node. Splitting the File into the N parts using blowfish sharing keys. Encrypt the file using the Key value. Provide the Time Stamp (TTL) for the File. File Downloading Process will send the download request to the server. If the file available in the network accept the download request. Then the client can downloading the file And Merge it.

C. Active Storage Object

Using the active storage object we can access the Storage node. For this one first we receive the File split from the user application. Get the Time stamp for the user application file. Create the interface. Store the files to the storage node and apply the security mechanism.

D. Self-Destruction Model with MAC

Time monitoring process will held. If the time stamp exceed the document will be automatically destroyed. The storage node doesn't get permission from the client to delete the data. The Store space will be refreshed. Details of the document also removed from the Meta server. Because of this the attackers can't get the details of the data.

MAC (Message Authentication code) In cryptography, a message authentication code (often MAC) is a short piece of information used to authenticate a message and to provide integrity and authenticity assurances on the message. Integrity assurances detect accidental and intentional message changes, while authenticity assurances affirm the message's origin.

V. 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 lever-aging of the essential properties of active storage framework based on T10 OSD standard. We demonstrated the feasibility of our approach by presenting time constrained, a proof- of-concept prototype based on object-based storage techniques. SeDas causes sensitive information, such as account numbers, passwords and notes to irreversibly self-destruct, without any action on the user's part. Our measurement and experimental security analysis sheds insight into the practicability of our approach. Our plan to re-lease the current SeDas system will help to provide researchers with further valuable experience to inform future object-based storage system designs for Cloud services.

REFERENCES

[1] Acharya M, Uysal and Saltz J, "Active disks: Programming model, algorithms and evaluation", in Proceedings 8th Conference Architectural Support for Programming Languages and Operating System (AS- PLOS), pp 81–91, October 1998.

[2] Geambasu R, Kohno T, Levy A, and Levy H M, "Vanish Increasing data privacy with self-destructing data", in Proceedings USENIX Security System Montreal, Canada, pp 299–315, August 2009.

[3] Wickremesinghe R, Chase J, and Vitter J, "Distributed computing with load-managed active storage", in Proceedings 11th IEEE International Conference High Performance Distributed Computing (HPDC), pp 13–23,2002.

[4] Weil S A, Brandt S A, Miller E L, Long D D E, and Malt Zahn C, "Ceph: A scalable, high-performance distributed file system", in Proceedings 7th Symp Operating Systems Design and Implementation (OSDI),2006.

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

[6] Wolchok S, Hofmann O S, Heninger N, Felten E W, Halderman J E, Rossbach C H, Waters B, and Witchel E, "Defeating vanishwith low-cost sybil attacks against large DHEs", in Proceedings Network and Distributed System Security, September 2010.

[7] Son S W, Lang S, Carns P, Ross R, Thakur R, Ozisikyilmaz B, "Enabling active storage on parallel I/O software stacks", in Proceedings IEEE 26th September Mass Storage Systems and Technologies (MSST), 2010.

[8] John T M, Ramani A T, and Chandy J A, "Active storage using object-based devices", in Proceedings IEEE International Conference, Cluster Computing , pp 472–478, 2008.

[9] Dimakopoulos V, Kinalis A, Mastrogiannakis S, and Pitoura E, "The smart autonomous storage (SMAS) system", in Proceedings IEEE Pacific Rim Conference.

[10] Ma X and Reddy A, "MVSS: An active storage architecture", IEEE Transaction on Parallel Distributed System volume 14, no. 10, pp 993–1003, October 2003.

[11] Welch B, Unangst M, Abbasi Z, Gibson G, Mueller G, Small J, Zelenka J, and Zhou B, "Scalable performance of the panasas parallel file system", in Proceedings 6th USENIX Conference, File and Storage Technologies (FAST), 2008.

[12] Xie Y, Muniswamy Reddy K K, Feng D, Long D D E, Kang Y, Niu Z, and Tan Z, "Design and evaluation of oasis An active storage framework based on t10 osd standard", in Proceedings 27th IEEE September Massive Storage Systems and Technologies (MSST), 2011.

[13] Riedel E, Fallouts C, Gibson G, and Nagle D, "Active disks for large scale data Proceeding sing", IEEE Computer, volume 34, no. 6, pp 68–74, June 2001.

[14] Chockler G and Malkhi D, "Active disk paxos with infinitely many processes", in Process 21st Annu Symp. Principles of Distributed Computing, pp 78–87, 2002.

[15] Keeton K, Patterson D A , and Hellerstein J, "A case for intelligent disks (IDISKs)", Special Interest Group on Management Of Data Resarch, volume 27, no. 3.