Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.1375 – 1379

RESEARCH ARTICLE

A Self Destructing Data System Based on Active Storage Framework for Protecting Data Privacy from attackers UN agency

R.Rengasamy[1], V.Kumaresan[2], G.Guru Rani[3]

[1]PG Student, Department of CSE & N.P.R College of Engg and Tech, Dindigul, Tamil Nadu, India

[2]PG Student, Department of CSE & N.P.R College of Engg and Tech, Dindigul, Tamil Nadu, India

[3]Assistant professor, Department of CSE & N.P.R College of Engg and Tech, Dindigul, Tamil Nadu, India

[1] rengasamy31@gmail.com; [2] kumaresanv2412@gmail.com; [3] ranitcguru@yahoo.com

Abstract— ***Personal knowledge hold on within the Cloud could contain account numbers, passwords, to be different necessary info that would be used and misused, a contender, or a court of law. These knowledge area unit cached, and archived by Cloud Service suppliers (CSPs), typically while not authorization and management. Self-destructing knowledge in the main aims at protective the user data's privacy. Data and data copies become destructed or unclear when a user-specified time, with none user intervention. Additionally, the secret writing key is destructed when the user-specified time. During this paper, we have a tendency to gift SeDas, a system challenge through a unique integration of scientific discipline with active techniques storage techniques supported T10 OSD commonplace. We have a tendency to enforce a proof-of-concept SeDas image. Through practicality and evaluations of the SeDas to meets all the privacy-preserving goals represented. System to be compared while not self-destructing knowledge mechanism, output for downloading with the projected SeDas tolerably decreases by but seventy two, whereas upload/download self-destructing with operations knowledge mechanism will increase by but hour.***

Keywords— ***Cloud Service Suppliers; Self Destructing Data System Based on Active Storage (SeDas); Distributed Hash Tables (DHT); Solid State Drive (SSD); Active Storage Object (ASO)***

I. INTRODUCTION

OUR development of Cloud computing and popularization of mobile web, Cloud services have become more and a lot of vital for people's life. Individual's square measure a lot of or less requested to submit or post some personal non-public info to the Cloud by the net. When individuals do that, they subjectively hope service suppliers can offer security policy to protect their knowledge from unseaworthy, therefore others individuals won't invade their privacy.

As folks believe additional and additional on the web and Cloud technology, security of their privacy takes additional and additional risks. On the one hand, once knowledge is being processed, reworked and stored by this ADPS or network, systems or network should cache, copy or archive it. These copies square measure essential for systems and also the network. However, folks haven't any knowledge regarding these copies and can't management them, so these copies might leak their privacy. On the opposite hand, their privacy also can be

leaked via Cloud Service suppliers (CSPs') negligence, hackers' intrusion or some legal actions. These issues present formidable challenges to shield people's privacy.

A pioneering study of Vanish [1] provides a replacement plan for sharing and protective privacy. Within the Vanish system, a secret key is divided and keep during a P2P system with distributed hash tables (DHTs). With connection and exiting of the P2P node, the system will maintain secret keys. Per characteristics of P2P, when concerning eight hours the DHT can refresh each node. With Shamir Secret Sharing algorithmic program [2], once one cannot get enough elements of a key, he won't decipher knowledge encrypted with this key, which suggests the secret is destroyed.

1) We have a tendency to specialize in the connected key distribution algorithmic rule, Shamir's algorithmic rule ,MD5 Algorithm[2], that is employed because the core algorithmic rule to implement shopper (users) distributing keys within the object storage system. We use these strategies to implement a safety destruct with equal divided key (Shamir Secret Shares [2]).

2) CloudME supported active storage framework, we have a tendency to use associate degree object-based storage interface to store and manage the equally divided key. We enforced a proof-of-concept SeDas epitome.

3) Through practicality and security properties analysis of the SeDas epitome, the results demonstrate that SeDas was sensible to use and meets all the privacy-preserving goals. The epitome system imposes fairly low run time overhead.

4) SeDas supports security erasing files and random cryptography keys hold on in an exceedingly magnetic disc drive (HDD) or solid state drive (SSD), severally. The rest of this paper is organized as follows. We have a tendency to review the related add Section II. We have a tendency to describe the design, design and implementation of SeDas in Section III. The intensive evaluations are given in Section IV, and that we conclude this paper in Section V.

II. RELATED WORK

Computing and communicating through the Web makes it virtually impossible to leave the past behind. A lost or stolen laptop can expose personal photos or messages; or a legal investigation can subpoena the entire contents of a home or work computer, uncovering just embarrassing details from the past. Our research seeks to protect the privacy of past, archived data such as copies of emails maintained by an email provider against accidental, the legal attacks. Specifically, we wish to all copies of certain data become unreadable after a user-specified time, without any specific action a user, without needing to trust any single third party to perform the deletion; attacker obtains both a cached copy of that data and the user's cryptographic keys and passwords. Vanish is a research project aimed at meeting this challenge through a novel integration of cryptographic techniques with distributed systems. We are implemented a proof-of-concept Vanish prototype that uses the million-node Vuze Bit Torrent DHT to create self-destructing data. The description of our Vuze-based self-destructing data system, please refer to our paper. We found that the initial Vuze DHT implementation on which Vanish was based was not adequately protected against Sybil attacks that seek to harvest data from the DHT. Due to overly eager replication for availability, part it is due to the fact that existing DHTs were not designed with such attacks in mind. We have been working with Paul Gardner to implement, and evaluate at scale measures for improving Vuze's security against Sybil-driven data-harvesting attacks. Our measures: (1) limit the excessive amount of replication that currently exists in Vuze, and (2) limit the ability of an attacker to perform large-scale Sybil attacks. Our evaluation shows combined defences significantly raise the bar against Sybil attacks. A comprehensive evaluation of all of our defences is currently underway and will be available shortly. In addition, we are investigating new directions and architectures for self-destructing data. The future for self-destructing data is to leverage multiple back-end storage systems (both DHTs and other types of distributed structures) in such a way that compromising Vanish would require compromising the entire storage node. As a proof of concept of this idea. We released a new prototype that splits the keys across both Vuze DHT and collaboration with Vinnie Moscaritolo from we are now investigating new storage backend for Vanish that have fundamentally different properties and threat models. a new developments in self-destructing data are underway, so stay tuned -- we will describe the latest advances in Vanish research on our publications page as they become available.

III. DESIGN AND IMPLEMENTATION OF SEDAS

Fig.1 shows the design of SeDas. There are 3 measure parties based on the active storage framework. i) Information server (MDS): MDS is user for chargeable management; server management and session management get into file information management. ii) Application node: the applying node could be a consumer

to use storage service of the SeDas. iii) Storage node: every storage node is Associate in Nursing OSD. It contains 2 core subsystems: key price store system and active storage object (ASO) runtime system. The key price store system that's supported the object storage element is employed for managing objects hold on in storage node: operation object, read/write object and then on. The object ID is employed as a key. The associated information and attribute square measure stored as values. The ASO runtime system supported the active storage agent module within the object-based storage system is employed to process active storage request from users and manage technique objects and policy objects.

A. Active Storage Object

Based on active storage framework, we have a tendency to use AN object-based storage interface to store and manage the equally divided key. We have a tendency to enforce a proof-of-concept Sedas image through practicality and security properties analysis of the Sedas prototype, the results demonstrate that Sedas is sensible to use and meets all the privacy-preserving goals. The image system imposes moderately low runtime overhead. The emergence of object-based interface, storage devices will benefit of the communicatory interface to realize some cooperation between application servers and storage devices.

B. Self Destructed Data

No express delete actions by the user, or any third-party storing that information No have to be compelled to modify any of the hold on or archived copies of that data. The time of key attainment is decided by DHT system and not governable for the user. This paper proposes a distributed object-based storage system with self-destructing information perform. In depth experiments show that the planned SeDas doesn't have an effect on the conventional use of storage system and might meet the wants of self-destructing information beneath a survival time by user governable key.

C. Data Process

A service methodology wants an extended time to method an advanced task, thus implementing code of a service methodology in user area will profit of performance of the system. All the info and their copies become destructed or indecipherable once a user-specified time. A destruct methodology object that's related to every secret key half and survival time parameter for every secret key half. The application node may be a shopper to use storage service of the SeDas. User's applications ought to implement logic of knowledge method and act as a shopper node. Once a user uploads a file to a storage system and stores his key during this SeDas system, he ought to specify the file. Any user United Nations agency has relevant permission will transfer information keep within the information storage system.

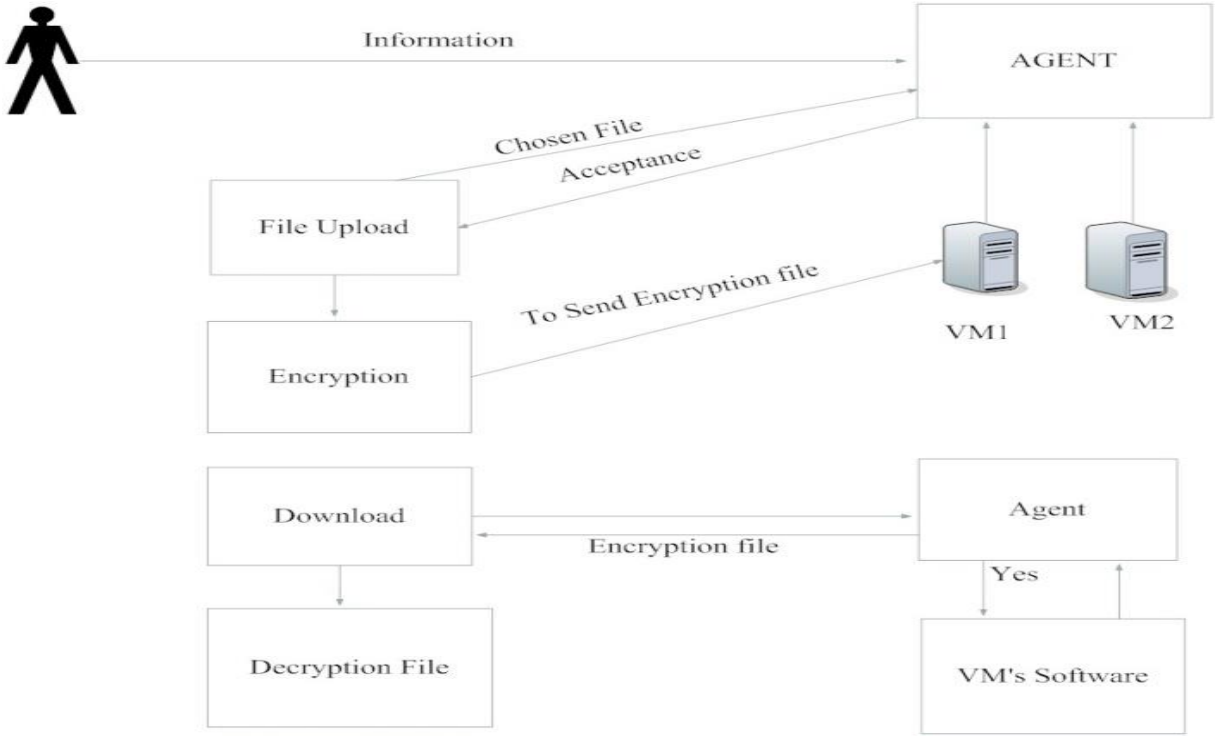

Fig. 1. SeDas system architecture

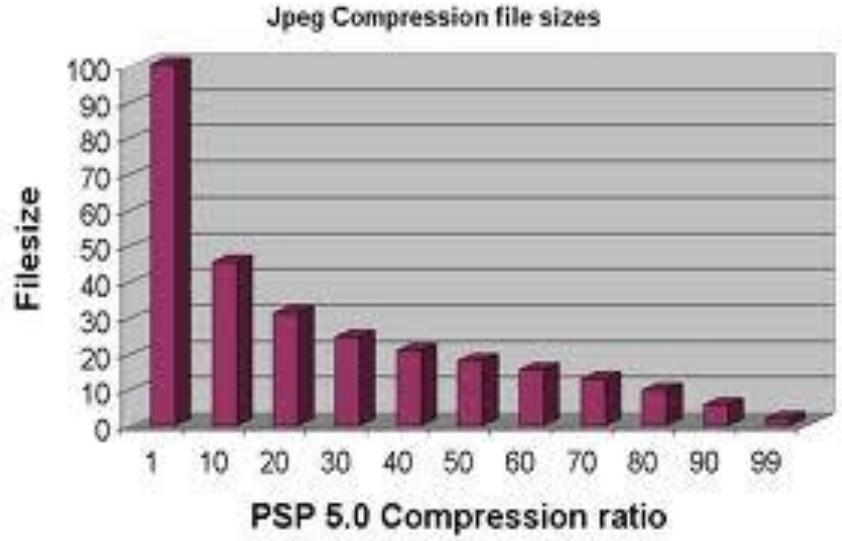

Fig. 2. Comparisons of latency in the upload and download operations.

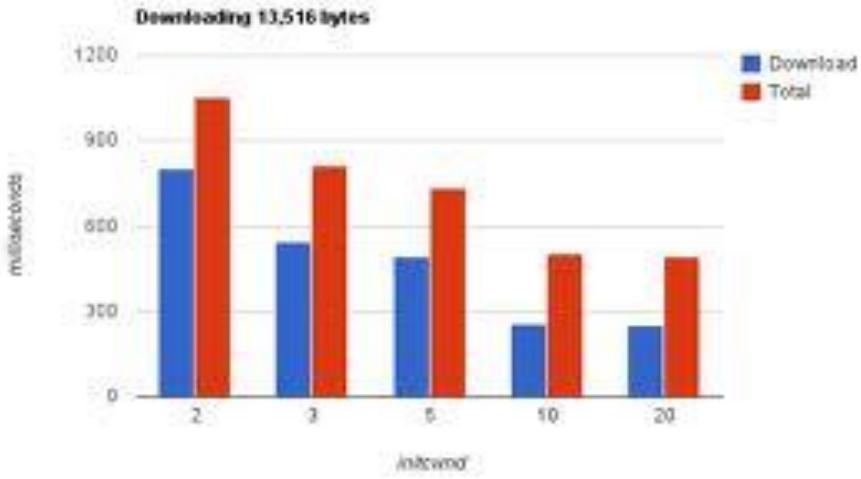

Fig. 3. Comparisons of overhead for encryption and decryption.

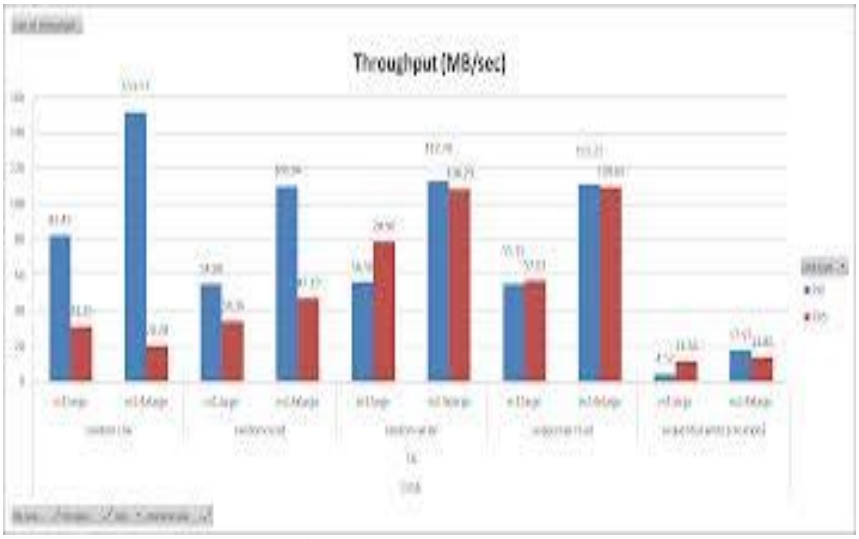

Fig. 4. Comparisons of throughput in the upload and download operations

IV. CONCLUSIONS

Data privacy has become progressively vital within the Cloud environment. This paper introduced a brand new approach for protecting data privacy from attackers UN agency retroactively get, through legal or different means that, a user's keep knowledge and private decryption keys. A completely unique side of our approach is that the leveraging of the essential properties of active storage framework based on T10OSD normal. We tend to incontestable the practicableness of our approach by presenting SeDas, a proof-of-concept prototype based on object-based storage techniques. SeDas causes sensitive information, like account numbers, passwords and notes to irreversibly destroy, with none action on the user's part .Our measurement and experimental security analysis into the practicableness of our approach. Our arrange to release the current SeDas system can facilitate to produce researchers with any valuable expertise to tell future object-based storage system styles for Cloud services.

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] 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] 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.
[5] 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.
[6] 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.
[7] T. M. John, A. Riya, and J. A. Chandy, "Active storage using object-based devices," in Proc. IEEE Int. Conf. Cluster Computing,2008, pp. 472–478.
[8] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service, "RFC 2212, IETF, Sept. 1997
[9] M. Carlson, W. Weiss, S. Blake, Z. Wang, D.Black, and E. Davies. "An Architecture for Differentiated Services," RFC 2475, IETF, Dec.1998
[10] R. Guerin and V. Peris, "Quality-of-service in packet networks: basic mechanisms and directions," Computer Networks Journal, Vol.31, No. 3, Feb. 1999
[11] Z. Dimitrijevic and R. Rangaswami, "Quality of service support for real-time storage systems, "in Proc. of Intl. IPSI-2003 Conference, (Stefan, Montenegro), October 2003
[12] K. Kim, J. Hwang, S. Lim, J. Cho, and K. Park, "A real-time disk scheduler for multimedia integrated server considering the disk internal scheduler," in Proc. of the International Parallel and Distributed Processing Symposium, Apr. 2003
[13] S. Brandt, S. Banachowski, C. Lin, and T.Bisson. "Dynamic integrated scheduling of hard real-time, soft real-time and non-real-time processes," in Proc. of the IEEE Real-Time Systems Symposium (RTSS '03), Dec. 2003.
[14] C. R. Lumb, A. Mrchant, G. A. Alvarez, Façade: virtual storage devices with performance guarantees, In Conference on File and Storage Technology (FAST 03), (San Francisco, CA),Mar. 2003
[15] E. Riedel, "Active disks - remote execution for network-attached storage,"Ph.D. dissertation, Electrical and Computer Engineering, Carnegie Mellon University, 1999, tech. Report no. CMU-CS-99-177.
[16] Information Technology - SCSI Architecture Model - 3 (SAM-3), ANSI, Sep. 2004.
[17] D. Du, D. He, C. Hong, J. Jeong, V. Kher, and Y. Kim, "Experiences in building an object-based storage system based on the OSD T-10 standard, "Digital Technology Center, University of Minnesota, Minneapolis, MN, Tech. Rep. DTC 2006/13, 2006.
[18] S. Y. W. Yu and G. J. Lipvski, "CASSM: A cellular system for very large data bases," in Proceedings of International Conference on Very Large Data Bases (VLDB), 1975.
[19] E. A. Ozharahan, S. A. Schuster, and K. C. Smith, "RAP: Associative processor for database management," in Proceedings of AFIPS Conference, 1975.

