ISSN: 2320-1363

Dynamic Group Based Secured Data Cluster Sharing in the Cloud Node

Malim Asma*1, K.Mahesh*2

M.Tech Scholar, Dept of CSE, Jayaprakash Narayan College of Engineering, Dharmapur, Mahabubnagar, AP, Inida

Assistant Professor, Dept of CSE, Jayaprakash Narayan College of Engineering, Dharmapur, Mahabubnagar, AP, India

ABSTRACT:

Technology and its significance in today's global village play the most important role. In the context of the taking consideration to Industry of Information Technology, we played the role making work structure and life style much easier where data Privacy is the typical factor to analyses dynamic group based solution. In this paper we have put forward the concept of the preserving the privacy in the so cause of the node based data node cluster where replication of the cloud node is much typical comparing to the existing orclassical approach. We have taken consideration of the access control based approach where data sharing would base on the group based specific and acknowledgment based approach to implement the best of the forward approach. In order to secure the best of the privacy we have implemented the dynamic group based encryption and decryption mechanism to some authorized group based on the clustered node.

KEYWORDS: Cloud computing, data sharing, privacy-preserving, access control, dynamic groups.

I.INTRODUCTION

In the Present of era of technology of the document sharing; The Document Object Model (DOM) is the representation of the base HTML page which is made available to JavaScript. It allows JavaScript to read and modify the base HTML content, as well as content referred to by the base HTML page. It also allows JavaScript to associate event handlers with events that are triggered by the user interacting with the HTML elements. We choose not to tie the JavaScript same origin policy to Document Object Model (DOM) access in this paper because it is possible to load content from a different origin without inserting it into the DOM. The JavaScript same origin policy restricts access to objects loaded by the web browser, regardless of whether they have been inserted into the DOM. In implementing SOMA, we extend policy dictating the fetching of content beyond JavaScript, going beyond the DOM interface. To verify the viability of the bin-locking proposal, we modified a 4.0 Linux system to implement bin-locking, including the kernel interface restrictions discussed in Chapter 4. The prototype implementation is composed of a number of different pieces which together protect the system. We wrote a binary signing utility which is used along with associated custom scripts to sign the binaries in the software archive, creating a new local mirror which we used for testing. We then installed these binaries on a test system using the package manager, which we modified to support bin-locked binaries. The Linux kernel on the test system was modified to enforce the proposed protection mechanisms (which include restrictions on bin-locked binaries as well as access to the kernel and file-systems). The boot process was modified on the test system to initialize kernel data structures which limit raw writes and mounting. We discuss each of these steps in detail below.

II.RELATED WORK

To increase control, we argue that extensibility and customizability should be built into public clouds. Specifically, clouds such as Amazon S3, Google Docs, Face book, and the DHT should allow their clients to customize data management properties, such as data placement, availability of forensic logs, and replication schemes. This chapter takes a first step toward the realization of the extensible-cloud vision, proposing the design, implementation, and evaluation of Comet, an extensible distributed storage system. While motivated broadly by the inflexibility in today's Web clouds, Comet focuses on a particular kind of cloud service – a distributed key/value store based on peer-to-peer DHTs. Comet's design is informed by our experience building Vanish on top of the inflexible DHT. We begin by describing this context and motivate the need for extensible DHTs.

By limiting the ability to read and modify content tagged with a different origin, many web attacks are prevented. The same origin policy prevents an attacker from performing the following attack Load an HTML page from a different origin. Parse the contents. Craft a subsequent request to the other origin based on the parse result. This restriction is important, as it blocks an attacker from implementing in JavaScript any multi-step attack which relies on the result of a previous request to any web server (other than that with the same origin as the JavaScript) in generating the next request to that server. The results of a request cannot be read (and hence parsed) in JavaScript if they came from a domain other

ISSN: 2320-1363

than the origin of the base HTML page; therefore the information required to make a subsequent request based on the first results is not made available to the attacker's JavaScript.

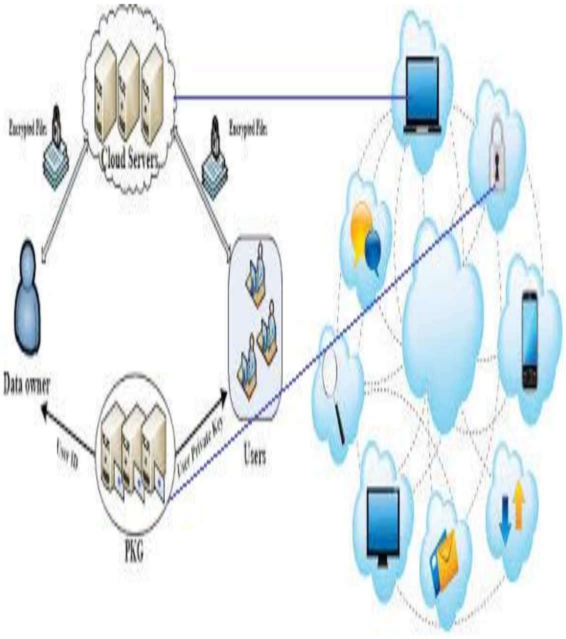

Fig.2.1 Illustration of Group Based Cloud Cluster Node

III.PROPOSED METHODOLOGY

After the motivation and an introduction of the Personal Secure Cloud a scenario shows how the ability to benefit from cloud's scalability without losing the data sovereignty enriches daily life. Afterwards the concept of my points out problems to be solved, lists the extracted research questions and my solution approach. Then related work is discussed followed by a short summary and outlook. What is cloud computing? There is no unique definition that everybody agrees on. A lot of definitions for example the one by NIST are complex. Cloud computing is nothing completely new. It is a combination of long existing technologies. In short, cloud computing can be understood as a distributed service approach which involves virtualization of physical servers and their rental, or the rental of services running on them. Because the focus is on data security, only storage services typically, system devices are composed of hardware and firmware working together to provide functionality. The hardware hooks into the system bus and exposes the functionality provided by the device (e.g., for an optical drive, this allows the software stack to access certain properties of the drive as well as the data on any inserted optical disk). The privileges of the device firmware, including its ability to interact with the rest of the system, are dictated by the underlying hardware (both that ofthe device the firmware is run on, as well as hardware the device is connected to). These restrictions can be in the form of specific hardware limits imposed to prevent damage or limits in the form of functionality which is simply not made available to the firmware. As an example, the range of motion for the read head for an optical drive is limited by the firmware, while the ability to prevent writing to optical media may be prevented

ISSN: 2320-1363

by simply installing a laser not powerful enough to actually 'burn' media.

The paging subsystem presents a translation layer between the virtual addresses used by software running on the processor and the physical addresses corresponding to the actual location in memory that is being referenced. Each block of virtual memory can be mapped to an arbitrary block of physical memory, with the exact mapping being maintained as an entry in the page table. The processor restricts the ability to update the page table to only privileged code (i.e., the OS kernel). The OS kernel is responsible for maintaining this mapping (for the moment, we will assume a hypervisor is not present). When an application is being run, the page table mappings are configured to allow access only to physical memory allocated to the currently running application. Because the mapping is controlled by the operating system, the application is restricted from accessing memory belonging to other applications on the system. While the OS kernel shares the same page table as the application, memory associated with the kernel remains protected through a privilege level bit enforced by the processor.

In its simplest form, the privilege level subsystem of a modern processor is a single bit, indicating whether the currently executing instructions have privileged or user level control of the processor [43]. Privileged mode is normally associated with the operating system kernel, with user mode being associated with all other code which is running. This separation between privileged and user mode is not the same as is commonly referred to in many access control systems.

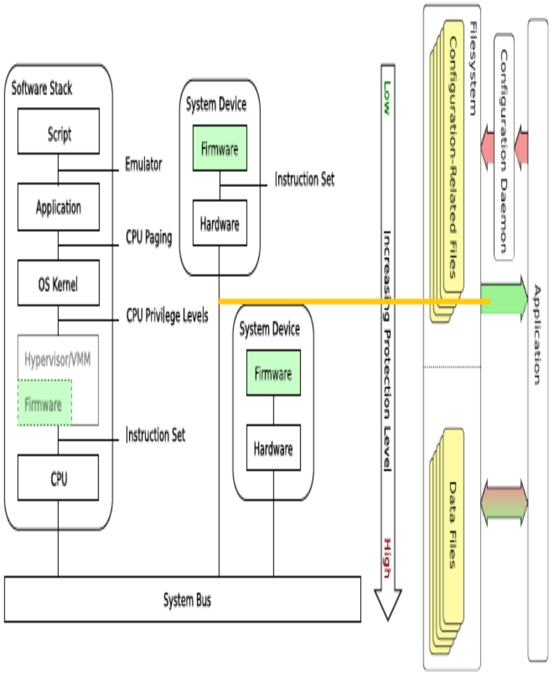

Fig.3.1.1 Cloud Architecture Model View of the Acknowledgement Based Firmware

IV.EVALUATION AND ANALYSIS

In addition to some assembly instructions only being accessible to code running with privileged control, the paging subsystem is capable of restricting access to memory based on whether the code currently running is privileged. A specific page of memory can be indicated as read-only for any user code running but read/write for privileged code. The OS kernel uses this privileged bit and associated access restrictions in the page tables to prevent user code from writing to pages belonging to the operating system.

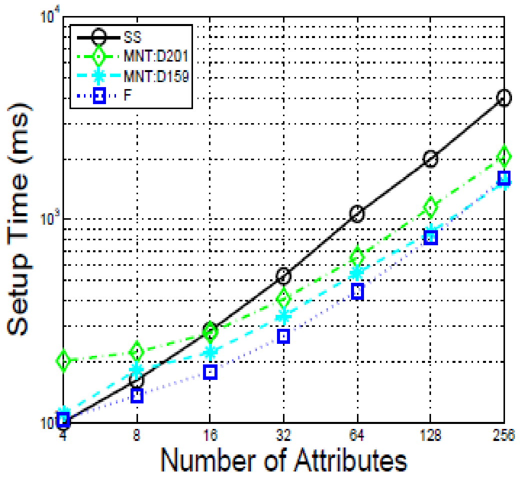

Fig.3.2.1 Evaluation Graph of Attribute and Time

Any attempt by user code to either execute privileged assembly instructions or modify read-only pages of memory is trapped by the processor and forwarded to the privileged code. The privileged code can either reject the attempt or emulate the operation on behalf of the user code.

V.CONCLUSION AND FUTUREWORK

This dissertation proposed a set of novel techniques to address specific data security, privacy, and management challenges raised by the adoption of new cloud and mobile technologies. The overarching goal of these techniques was to increase users' control over various aspects of their data in the cloud and on mobile devices. Keypad provides remote access auditing and control over data stored on stolen mobile devices; Vanish offers data lifetime control on the Web; Comet allows users to customize various data management properties in a storage cloud; and Menagerie allows users to regain a unified organizational view over their scattered Web data. These systems show that, with carefully crafted abstractions and mechanisms, users can regain control over various aspects of their data without losing the new technologies' advantages.

VI.REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[2] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc.Int'l Conf. Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010.

[3] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM, pp. 534-542, 2010.

[4] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable Secure File Sharing on Untrusted Storage," Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[5] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing Remote Untrusted Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.

[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2005.

[7] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[8] B. Waters, "Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," Proc. Int'l Conf. Practice and Theory in Public Key Cryptography Conf. Public Key Cryptography, http://eprint.iacr.org/2008/290.pdf, 2008.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. ACM Conf. Computer and Comm. Security (CCS), pp. 89-98, 2006.

[10] D. Naor, M. Naor, and J.B. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Proc. Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001.

[11] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 213-229, 2001.

[12] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signature," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-55, 2004.

[13] D. Boneh, X. Boyen, and E. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," Proc. Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 440-456, 2005.

[14] C. Delerablee, P. Paillier, and D. Pointcheval, "Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys," Proc. First Int'l Conf. Pairing-Based Cryptography, pp. 39-59, 2007.

[15] D. Chaum and E. van Heyst, "Group Signatures," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), pp. 257-265, 1991.