SECURE OVERLAY CLOUD STORAGE WITH ACCESS CONTROL AND ASSURE DELETION

[1]A.Mohammed Shaffi,[2] C.Sivakumar

[1]PG Scholar, Department of Information Technology, Jayam College of Engineering and Technology, Dharmapuri

[2]Associate Professor/IT, Department of Information Technology, Jayam College of Engineering and Technology, Dharmapuri

[1]mohd.shaffu@gmail.com, [2]svkumar650@gmail.com

ABSTRACT: In proposed system user outsource data backups off-site to third-party cloud storage services so as to reduce data management costs. However, user must provide security guarantees for the outsourced data, which is now maintained by third parties. A design and implement FADE, and a secure overlay cloud storage system that achieves fine-grained, policy- based access control and file assured deletion. It associates outsourced files with file access policies, and assuredly deletes files to make them unrecoverable to anyone upon revocations of file access policies. To achieve such security goals, FADE is built upon a set of cryptographic key operations that are self-maintained by a quorum of key managers that are independent of third-party clouds. In particular, FADE acts as an overlay system that works seamlessly atop today's cloud storage services. Implement a proof-of-concept prototype of FADE atop Amazon S3, one of today's cloud storage services. By conduct extensive empirical studies, and demonstrate that FADE provides security protection for outsourced data, while introducing only minimal performance and monetary cost overhead. This work provides, value added security features incorporate were today's cloud storage service.

Keywords: Cloud Storage, fine grained, policy based access control.

1. INTRODUCTION

Cloud storage is a new business solution for remote backup outsourcing, as it offers an abstraction of infinite storage space for clients to host data backups in a pay-as you- go manner. It helps enterprises and government agencies significantly reduce their financial overhead of data management, since they can now archive their data backups remotely to third-party cloud storage providers rather than maintain data centers on their own. For example, Smug Mug, a photo sharing website, chose to host terabytes of photos on Amazon S3 in 2006 and saved thousands of dollars on maintaining storage devices.

More case studies of using cloud storage for remote backup can be found in. Apart from enterprises and government agencies, individuals can also archive their personal data to the cloud using tools like Drop box. In particular, with the advent of smart phones, we expect that more people will use Drop box-like tools to move audio/video files from their smart phones to the cloud, given that smart phones typically have limited storage resources.

However, security concerns become relevant as we now outsource the storage of possibly sensitive data to third parties. In this paper, we are particularly interested in two security issues.

First, we need to provide guarantees of access control, in which we must ensure that only authorized parties can access the outsourced data on the cloud. In particular, we must prohibit third-party cloud storage providers from mining any sensitive information of their clients' data for their own marketing purposes.

Second, it is important to provide guarantees of assured deletion, meaning that outsourced data is permanently inaccessible to anybody (including the data owner) upon requests of deletion of data. Keeping data permanently is undesirable, as data may be unexpectedly disclosed in the future due to malicious attacks on the cloud or careless management of cloud operators.

The challenge of achieving assured deletion is that we have to trust cloud storage providers to actually delete data, but they may be reluctant in doing so. Also, cloud storage providers typically keep multiple backup copies of data for fault-tolerance reasons. It is uncertain, from cloud clients' perspectives, whether

cloud providers reliably remove all backup copies upon requests of deletion. The security concerns motivate us, as cloud clients, to have a system that can enforce access control and assured deletion of outsourced data on the cloud in a fine-grained manner. However, building such a system is a difficult task, especially when it involves protocol or hardware changes in cloud storage infrastructures that are externally owned and managed by third-party cloud providers. Thus, it is necessary to design a secure overlay cloud storage system that can be overlaid and work seamlessly atop existing cloud storage services.

In this paper, we present FADE, a secure overlay cloud storage system that provides fine grained access control and assured deletion for outsourced data on the cloud, while working seamlessly atop today's cloud storage services. In FADE,active data files that remain on the cloud are associated with a set of user-defined file access policies (e.g., time expiration, read/write permissions of authorized users),such that data files are accessible only to users who satisfy the file access policies.

In addition, FADE generalizes time-based file assured deletion (i.e., data files are assuredly deleted upon time expiration) into a more fine-grained approach called policy- based file assured deletion, in which data files are assuredly deleted when the associated file access policies are revoked and become obsolete.

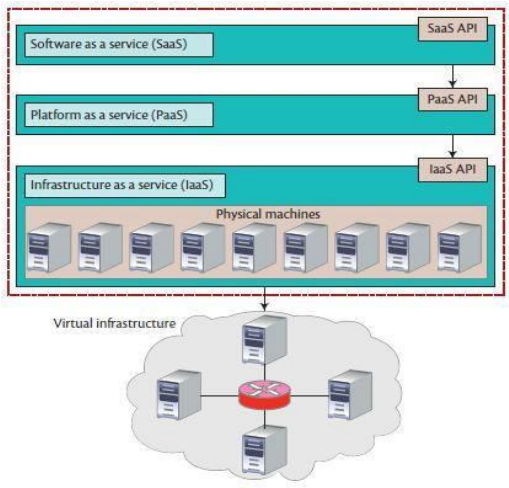

Figure1: System Architecture

2. RELATED WORK ON CLOUD SECURITY AND ACCESS CONTROL

Cloud storage is a new business solution for remote backup outsourcing, as it offers an abstraction of infinite storage space for clients to host data backups in a pay-as you-go manner. Time based File assured Deletion is the Existing approach. Time-based file assured deletion, which is first introduced in, means that files can be securely deleted and remain permanently inaccessible after a pre-defined duration. The main idea is that a file is encrypted with a data key by the owner of the file, and this data key is further encrypted with a control key by a separate key manager. The key manager is a server that is responsible for cryptographic key management. The control key is time- based, meaning that it will be completely removed by the key manager when an expiration time is reached, where the expiration time is specified when the file is first declared. Without the control key, the data key and hence the data file remain encrypted and are deemed to be inaccessible. Thus, the main security property of file assured deletion is that even if a cloud provider does not remove expired file copies from its storage, those files remain encrypted and unrecoverable.

Later, the idea of time-based file assured deletion is prototyped in Vanish. Vanish divides a data key into multiple key shares, which are then stored in different nodes of a public Peer-to-Peer Distributed Hash Table (P2P DHT) system.

3. FADE

In this paper, we are particularly interested in two security issues. First, we need to provide guarantees of access control, in which we must ensure that only authorized parties, can access the outsourced data on the cloud. In particular, we must prohibit third-party cloud storage providers from mining any sensitive information of their clients' data for their own marketing purposes. Second, it is important to provide guarantees of assured deletion, meaning that outsourced data is permanently inaccessible to anybody (including the data owner) upon requests of deletion of data

We now generalize time-based deletion to policy-based deletion as follows. We associate each file with a single atomic file access policy (or policy

for short), or more generally, a Boolean combination of atomic policies. Each (atomic) policy is associated with a control key, and all the control keys are maintained by the key manager. Suppose now that a file is associated with a single policy. Then similar to time-based deletion, the file content is encrypted with a data key, and the data key is further encrypted with the control key corresponding to the policy. When the policy is revoked, the corresponding control key will be removed from the key manager. Thus, when the policy associated with a file is revoked and no longer holds, the data key and hence the encrypted content of the file cannot be recovered with the control key of the policy. In this case, we say the file is assuredly deleted. The main idea of policy-based deletion is to delete files that are associated with revoked policies.

3.1 Login Process Denial of Services.

It may be possible to overwhelm the login process by continually sending login-requests that require the presentation tier to access the authentication mechanism, rendering it unavailable or unreasonably slow to respond.

When a user enters an incorrect username and/or password, the application should respond with a generic error message stating that the information entered was incorrect. If the application explicitly states which component of the username/password pair was incorrect then an attacker can automate the process of trying common usernames from a dictionary file in an attempt to enumerate the users of the application. Whilst applications may handle authentication failure messages correctly, many still allow attackers to enumerate users through the forgotten password feature.

3.2 Group Attacker Modules

The maximum destruction caused by the attacks includes the depletion of the application service resource at the server side, the unavailability of service access to legitimate user, and possible fatal system errors which require rebooting the server for recovery. We assume that any malicious behaviors can be discovered by monitoring the service resource usage, based on dynamic value thresholds over the monitored objects. Data manipulation and system intrusion are out of this scope. That application interface presented by the servers can be readily discovered and clients communicate with the servers using HTTP/1.1 sessions on TCP connections.

We consider a case that each client provides a non-spoofed ID, which is utilized to identify the client during our detection period. Despite that the application DoS attack is difficult to be traced; by identifying the IDs of attackers the firewall can block the subsequent malicious requests. The attackers are assumed to launch application service requests either at high inter arrival rate or high workload, or even both. The term "request" refers to either main request or embedded request for HTTP page. Since the detection scheme proposed will be orthogonal to the session affinity, we do not consider the repeated one-shot attack mentioned in. We further assume that the number of attackers d << n where n is the total client amount. This arises from the characteristics of this attack. Due to the benefits of virtual server s we employee, this constraint can be relaxed, but we keep it for the theoretical analysis in the current work.

3.3 Group Testing

The classic GT (Group Testing0 model consists of t pools and n items (including at most d positive ones). This model can be represented by a t _ n binary matrix M where rows represent the pools and columns represent the items. An entry M[I, j]= 1 if and only if the I th pool contains the j th item; otherwise, M[I, j]= 0. The t-dimensional binary column vector V denotes the test outcomes of these t pools, where 1-entry represents a positive outcome and 0-entry represents a negative one.

Note that a positive outcome Indicates that at least one positive item exists within this pool; whereas negative one means that all the items in the current pool are negative.

A detection model based on GT can be assume that there are t virtual servers and n clients, among which d clients are Binary testing matrix M and testing outcome vector V. Attackers. Consider the matrix M t*n in Fig. 1, the clients can be mapped into the columns and virtual servers into rows in M, where M[I, j]= 1 if and only if the requests from client j are distributed to virtual server i. With regard to the test outcome column V, we have V[i]= 1 if and only if virtual server i has received

malicious requests from at least one attacker, but we cannot identify the attackers at once unless this virtual server is handling only one client. Otherwise, if V ½i_ ¼ 0, all the clients assigned to server I are legitimate. The d attackers can then be captured by decoding the test outcome vector V and the matrix M. Victim/Detection

The victim model in our general framework consists of multiple back-end servers, which can be Web/application servers, database servers, and distributed file systems. We do not take classic multitier Web servers as the model, since our detection scheme is deployed directly on the victim tier and identifies the attacks targeting at the same victim tier; thus, multitier attacks should be separated into several classes to utilize this detection scheme. We assume that all the back-end servers provide multiple types of application services to clients using HTTP/1.1 protocol on TCP connections.

Each back-end server is assumed to have the same amount of resource. Moreover, the application services to clients are provided by K virtual private servers (K is an input parameter), which are embedded in the physical back-end server machine and operating in parallel. Each virtual server is assigned with equal amount of static service resources, e.g., CPU, storage, memory, and network bandwidth. The operation of any virtual server will not affect the other virtual servers in the same physical machine. There a sons for utilizing virtual servers are twofold. First, each virtual server can reboot independently, thus is feasible for recovery from possible fatal destruction; second, the state transfer overhead for moving clients among different virtual servers is much smaller than the transfer among physical server machines.

3.4 File Upload/Download

The client encrypts the input file according to the specified policy (or a Boolean combination of policies). Here, the file is encrypted using the 128-bit AES algorithm with the cipher block chaining (CBC) mode. After encryption, the client also appends the encrypted file size (8 bytes long) and the HMAC-SHA1 signature (20 bytes long) to the end of encrypted file for integrity checking in later downloads. It then sends the encrypted file and the metadata onto the cloud. The client retrieves the file and policy metadata from the cloud. It then checks the integrity of the encrypted file, and decrypts the file.

3.5 Policy Revocation and Renewal

The client tells the key managers to permanently revoke the specified policy. All files associated with the policy will be assuredly deleted. If a file is associated with the conjunctive policy combination that contains the revoked policy, then it will be assuredly deleted as well. The client first fetches the metadata of the given file from the cloud. It updates the metadata with the new policy. Finally, it sends the metadata back to the cloud. Note that the operation does not involve transfer of the input file.

3.6 Cloud Storage

The cloud, maintained by a third-party provider, provides storage space for hosting data files on behalf of different FADE clients in a pay-as-you-go manner. Each of the data files is associated with a combination of file access policies. FADE is built on the thin-cloud interface, and assumes only the basic cloud operations for uploading and downloading data files.

4. CONCLUSION

In This System propose a practical cloud storage system called FADE, which aims to provide access control assured deletion for files that are hosted by today's cloud storage services. Associate files with file access policies that control how files can be accessed. and then present policy-based file assured deletion, in which files are assuredly deleted and made unrecoverable by anyone when their associated file access policies are revoked. Describe the essential operations on cryptographic keys so as to achieve access control and assured deletion.

FADE also leverages existing cryptographic techniques, including attribute based encryption (ABE) and a quorum of key managers based on threshold secret sharing. implement a prototype of FADE to demonstrate its practicality, and empirically study its performance overhead when it works with Amazon S3.Propose experimental results provide insights into the performance security trade- off when FADE is deployed in practice.

REFERENCES

[1] Bertino.F, Mar.2009, "Privacy- Preserving Digital Identity Management for Cloud Computing,", IEEE Computer Society Data Engineering Bulletin pp. 1–4.

[2] Blaze et al. M,2009, "Dynamic Trust Management," Computer, vol. 42, no. 2, pp. 44–52.

[3] Definition of Cloud Computing, October, 2009. National Institute of Standards and Technology, Version 15.

[4] Duffy Marsan .C, 10 May 2011, "Verisign to Extend Cloud-Based DDoS Protection to SMEs," ComputerWorld UK, www.computerworlduk.com/news/security/32788 05/verisign-to-extend-cloud-ased-ddos - protectionto-smes.

[5] Gartner, February 17, 2011, Survey Shows U.S." Consumers More Likely to Purchase a Smartphone Than Other Consumer Devices in 2011", Gartner Press Release, Accessed 4, 7thJune2011.http://www.gartner.com/it/page.jsp?id=1550814.

[6] Hu .H, 2009 "Security-Enhanced OSGi Service Environments," IEEE Trans. Systems, Man, and Cybernetics-Part C: Applications and Reviews, vol. 39, no. 5, pp. 562–571.

[7] Joshi .J, 2004, "Access Control Language for Multidomain Environments," IEEE Internet Computing, vol. 8, no. 6, pp. 40–50.

[8] Joshi J.B.D, 2010 ,"SecureCloud: Towards a Comprehensive Security Framework for Cloud Computing Environments," Proc. 1st IEEE Int'lWorkshop Emerging Applications for Cloud Computing(CloudApp 2010), IEEE CS Press, , pp. 393–398.

[9] Joshi et al .J, 2010,"State Miner: An Efficient Similarity- Based Approach for Optimal Mining of Role Hierarchy," Proc. 15th ACM Symp. Access Control ModelsAnd Technologies, ACM Press, pp. 55–64.

[10] Paxson . V , 2010, "What's New About Cloud Computing Security?" tech. report UCB/EECS- 2010-5, EECS Dept., Univ. of California

[11] Perlner. R, 2010, "Why Aren't Cloud Services Secured as a Service?", blog,2009 [online] Available: http://cloudsecurity. Trendmicro.com/why-arent-cloud-services secured- as-a-service [3] DRM.In IDtrust Berkeley,www.eecs.berkeley.edu/Pubs/TechRpts/2010 /EECS-2010-5html.

[12] Shehab.M 2009,"Privacy -Enhanced User-Centric Identity Management," Proc.IEEE Int'l Conf. Communications IEEE Press , PP.998-1002.

[13] Shin .D, 2005, "Role-Based Privilege and Trust Management," Computer Systems Science & Eng. J., vol. 20, no. 6, , pp. 401–410.

[14] Takabi .H, 2010, "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security & Privacy, vol. 8, no. 6 pp . 24–31.

[15] Zhang .Y, 2009, "Access Control and Trust Management For Emerging Multidomain Environments," Annals of Emerging Research in Information Assurance, Security and Privacy Services, S. Upadhyaya and R.O. Rao, eds., Emerald Group Publishing, pp. 421–452.