International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 7, Issue 3, May - June 2018 ISSN 2278-6856

To incorporate value-added security features into current data for outsourcing cloud data applications through Secured Access Control and Assured Deletion

Mr. ShivaKumara T[1], Mr. Muneshwara M.S[2], Dr. Lokesh A[3], Dr. Rajashekar M Patil[4]

[1]Assistant Professor, Department of MCA, BMS Institute of Technology & Management, Bengaluru -560064, Karnataka, India.

,

[2]Assistant Professor, Department of CS&E, BMS Institute of Technology & Management, Bengaluru -560064, Karnataka, India.

[3]Associate Professor, Department of IS&E, Sri Venkateshwara College Of Engineering, Bengaluru, Karnataka, India.

[4]Professor, Bengaluru, Karnataka, India.

Abstract: ***Data can be outsourced as backups off-site to third-party cloud storage services in order to reduce data management costs. However, security must be provided for the outsourced data, which is maintained by third parties. We implement FADE, a secure cloud storage system that basically achieves fine-grained; file assured deletion and policy-based access control. It also associates outsourced files with file access policies and securely deletes files to make them unrecoverable to anyone upon revocations of file access policies. To achieve these security goals, FADE is based on a set of cryptographic key operations that are self-maintained by a quorum of key managers that are usually independent of third-party clouds. FADE acts as a secure overlay system that works seamlessly a top today's cloud storage services. To implement a working prototype of FADE atop Amazon S3, one of the most happening cloud storage services, and empirically shows that FADE provides policy-based file assured deletion with a minimal trade-off of performance overhead. Our work provides the information about how to incorporate value-added security features into current data outsourcing applications.***

Keywords: *Attribute- based encryption, Cryptography, FADE, Metadata, Policy-based access control, Secure overlay.*

1. INTODUCTION

Cloud computing is an emerging technology which is being used widely these days. Now a days more and more organizations are now using cloud to outsource their data for sharing due to the cost-effectiveness, flexibility and scalability of cloud. Cloud storage provides an abstraction

of infinite storage space for clients to host data, in a pay-as-you-go manner. For example, SmugMug, a photo sharing website, chose to host terabytes of photos on Amazon S3 in 2006 and saved about 500K US dollars on storage devices. Thus, instead of having own data centres, enterprises can now outsource the storage of a large amount of digitized content to those third-party cloud storage providers to save the financial overhead in data management. Apart from enterprises, individuals can also be benefitted from cloud storage as a result of the advent of mobile devices (e.g., smartphones, laptops). Given that mobile devices will be having limited storage space in general, individuals can send audio/video files to the cloud and can make effective use of space in their mobile devices.

2. RELATED WORK

However, privacy and integrity concerns are relevant as we now consider third parties to host possibly sensitive data. To protect our outsourced data, a straightforward approach is required. To apply cryptographic encryption onto sensitive data with a set of encryption keys, maintaining and protecting such encryption keys will create another security issue. One specific issue is that upon the requisition of deletion of files, cloud storage providers may not completely delete all file copies (e.g., cloud storage providers may make multiple file backup copies and distribute them over the cloud for reliability, and clients do not know the number or even the existence of these backup copies), and eventually disclose the data if the encryption keys are unexpectedly obtained, either by accidents or by malicious attacks. Therefore, we seek to achieve a major security goal called file assured deletion, meaning that files are reliably deleted and permanently unrecoverable and they remain inaccessible.

1) Privacy Preserving Public Auditing For Secure Cloud Storage

By Cong Wang, Student member, IEEE.2010, according to this Paper Doesn't Have Policy Based, Data key, Access Keys Are Not There, We Propose Random Masking Technique for Single Secrete Key Only.

2) Policy Based Access Control for Diverse Dod Security Domains

By Brad, CoxTechnica Corporation: 2011, PhD. The Data Into Temporarily. Timing Polices and Access Control Is Not There. The Data Key and Secrete Key And Access Key Three Keys Are Act As A Master Key.

3 SYSTEM ANALYSIS

Systems analysis is an issue solving technique that decomposes a system into its part pieces for the purpose of the studying how well those component parts work and interrelate to accomplish their purpose.

3.1 EXISTING SYSTEM

Time-based file assured deletion, introducing in the existing system which means that files can be securely deleted and can remain permanently inaccessible by anyone after a pre-defined duration of time. The main idea is that a file is encrypted with a data key only by the owner of the file, and again this data key is further encrypted with a control key by a separate key manager. The key manager is a server who 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 this expiration time will be specified when the file is first declared. Without the control key, the data key and hence the data file remain encrypted and are inaccessible.

DISADVANTAGES

Without the control key, the data key cannot be generated and hence the data file remain encrypted hence are deemed to be inaccessible.

The main security property of file assured deletion is that even if a cloud provider does not remove expired file copies from the storage, those files remain encrypted and unrecoverable and hence the files are inaccessible.

3.2 PROPOSED SYSTEM

We propose a secure cloud storage system which is called FADE, that aims to ensure the access control assured deletion for file that are hosted by today's cloud storage services. We associate these files with file access policies that control how files can be accessed based on the policies. We 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. We describe the essential operations. On cryptographic keys used 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. We implement a prototype of FADE to demonstrate its practicality, and empirically study its performance overhead when it works with Amazon S3. Our experimental results provide the information about the cloud backup, storage files data safely and securely.

ADVANTAGES

Cloud provides the security services to the end users.

4. IMPLEMENTATION

Now, we propose a cloud storage system called as FADE, which is looking forward to provide access control assured deletion for files that are hosted by today's cloud storage services. We associate files with file access policies that control how files can be accessed based on the policy. We 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.

We 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.

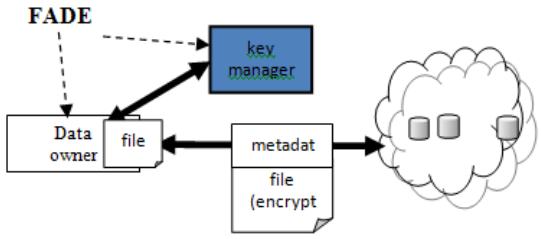


Figure 1: The FADE system

The prototype of FADE is implemented to demonstrate its practicality, and empirically study its performance overhead when it working with Amazon S3. Our experimental results provide the brief information about the performance-security trade-off when the FADE is deployed practically.

4. MATHEMATICAL MODULES

We now introduce the basic operations of how a client Uploads or downloads files to or from the cloud. We start with the case where each file is associated with a single policy

4.1 FILE UPLOAD:

System=<Input, Output, Process>
Input:
{
No. of Files to be processed
}

Output:
{
No. of files uploaded successfully
}
Process :{ Key generation}

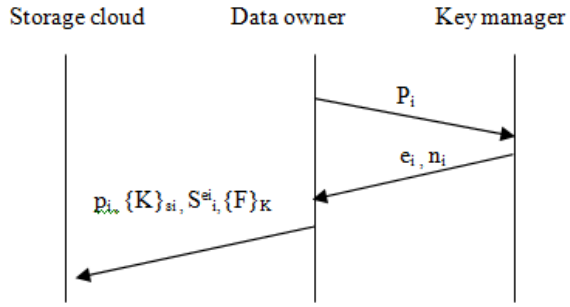
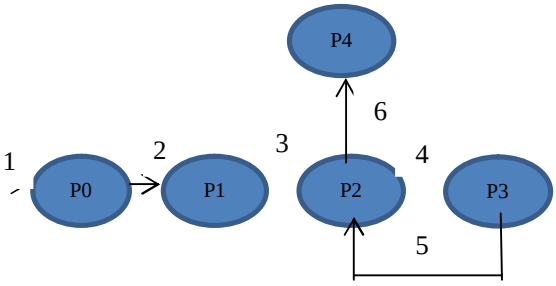


Figure 2: File upload operation.



1. User login
2. Select file
3. Key generation
4. Key manager
5. Key generation
6. Upload in cloud.

Figure 3: Turing Machine for File Upload Operation.

Here, the client first requests for the public control key (ni, ei) of policy Pi from the key manager, and catches (ni, ei) for subsequent uses if the same policy Pi is associated with other files. Then the client generates two random keys they are K and Si, and sends {K} Si, Seii and {F} K to the cloud2. Then the client must discard K and Si. To protect the integrity of a file, the client computes HMAC signature on every encrypted file and the HMAC signature will be stored together with the encrypted file in the cloud. We assume that the client has a long-term private secret value for the HMAC computation. Si, and decrypt {K} Si and hence {F} K.

4.2 FILE DOWNLOAD

System=<Input, Output, Process>
Input:
{
No. of Files to be processed
}
(Storage cloud
{
No. of files downloaded successfully
}
Process :{ Key generation}

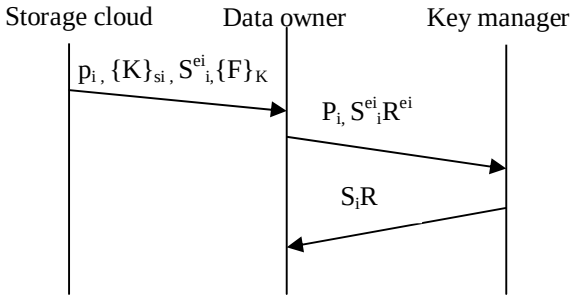
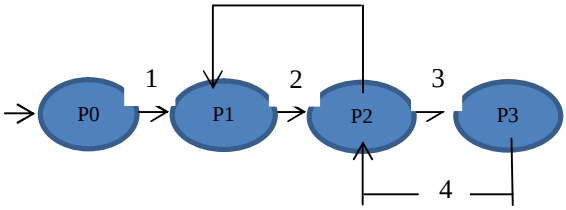


Figure 4: File download operation



1.request for download
2.get the key for decryption
3.key manager
4.decrypted key
5.download successful

Figure 5: Turing Machines for file download operation.

The client fetches {K} Si, Seii, and {F} K from the cloud. The client will first check for the validation of the HMAC signature before decrypting the file. Then the client will generate a secret random number R, computes Rei, and sends Seii ·Rei = (SiR) ei to the key manager for the requisition of decryption. The key manager then computes and returns ((SiR) ei) di = SiR to the client, which can now remove R and obtain Si, and decrypt {K} Si and hence {F} K.

5. EVALUATION

We now evaluate the empirical performance of our implemented prototype of FADE atop Amazon S3. It is crucial that FADE does not introduce substantial performance or monetary overhead that will lead to a big increase in data management costs. In addition, the cryptographic operations of FADE should only bring insignificant computational overhead. Therefore, our experiments aim to answer the following questions: What is the performance and monetary overhead of FADE? Is it feasible to use FADE to provide file assured deletion for

cloud storage? Our experiments use Amazon S3 APAC servers that reside in Singapore for our cloud storage backend. Also, we deploy the client and the key managers within a departmental network. We evaluate FADE on a per-file basis, that is, when it operates on an individual file of different sizes. We can proportionally scale our results for the case of multiple files.

5.1 TIME PERFORMANCE OF FADE

We first measure the time performance of implemented prototype of FADE. In order to identify the time overhead of FADE, we just divide the running time of each measurement into three components. They are

1. File transmission time, the uploading/downloading time for the data file between the client and the cloud.
2 .Metadata transmission time, the time for uploading/ downloading the metadata, which contains the policy information and the cryptographic keys associated with the file, between the client and the cloud.
3. Cryptographic operation time, the total time for cryptographic operations, which also includes the total computational time used for performing AES and HMAC on the file, and the time required by the client to Coordinate with the quorum of key managers on operating the cryptographic keys. We take the average of our measurement results over 10 different trials.

We evaluate the time performance of the basic design of DFADE, in which we use a single key manager and do not involve ABE.

Performance of file upload/download operations:

In this experiment, we measure the running time of the file upload and download operations for different file sizes (including 1KB, 3KB, 10KB, 30KB, 100KB, 300KB, 1MB, 3MB, and 10MB). First, the cryptographic operation time increases with the file size, mainly because of the symmetric-key encryption applied to a larger file. Nevertheless, we find that in all cases of file upload / download operations, the time of cryptographic operations is no more than 0.2s (for a file size within 10MB), and accounts for no more than 2.6% of the file transmission time. We then expect the FADE to introduce a small time overhead in cryptographic operations as compared to the file transmission time, where the latter is inevitable even without FADE.

7. CONCLUSION

We proposed a cloud storage system that is called FADE, which aims to provide assured deletion for files that are hosted by today's cloud storage services. We present the design of policy-based file assured deletion, in which files are assuredly deleted and made unrecoverable by anyone when their associated file access policies are revoked. Operations on cryptographic keys are ensured so as to achieve policy-based file assured deletion. We implement a prototype of FADE to demonstrate its practicality, and empirically study its performance overhead when it works with Amazon S3. Finally, our experimental results provide the depth of knowledge about the performance-security trade-off when FADE is deployed in practice.

ACKNOWLEDGMENT

The authors would like to thank the editor, mysterious reviewers for their valuable suggestions that appreciably improved the quality of this paper, specially thank to the Guide and HOD for their constant encouragement and providing us the overwhelming support and guidance to write this paper. Finally also thankful our dear colleagues and friends.

REFERENCES

[1] Yang Tang, Patrick P. C. Lee, John C. S. Lui, Radia Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion ", IEEE TRANSACTIONS AND DEPENDABLE AND SECURE COMPUTING VOL.9 NO.6 , 2012.

[2] Amazon Case Studies. http :// aws. amazon. Com / solutions/casestudies/#backup.

[3] Amazon. Smug Mug Case Study: Amazon Web Services. http:// aws.amazon.com /solutions/ case-studies /smugmug/, 2006.

[4] Amazon S3. http://aws.amazon.com/s3, 2010.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, Stoica, and M. Zaharia. A View of Cloud Computing. *Comm. of the ACM*, 53(4):50–58, Apr 2010.

[6] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik. Scalable and Efficient Provable Data Possession. In *Proc. Of SecureComm*, 2008.

[7] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-Policy Attribute-Based Encryption. In *Proc. of IEEE Symp. On Security and Privacy*, May 2006.

[8] A. Boldyreva, V. Goyal, and V. Kumar. Identity-based Encryption with Efficient Revocation. In *Proc. of ACM CCS*, 2008.

[9] T. Dierks and E. Rescorla. The transport layer security (tls) protocol version 1.2, Aug 2008. RFC 5246.

[10] Dropbox. http://www.dropbox.com, 2010.

[11] H. Abu-Libdeh L. Princehouse, and H. Weatherspoon, "RACS: A Case for Cloud Storage Diversity," Proc. ACM First ACM Symp. Cloud Computing (SoCC).

[12] Dropbox, http://www.dropbox.com.

[13] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246

[14] W. Stallings, Cryptography and Network Security. Prentice Hall, 2006.

[15] "Service Oriented Architecture & Web Services". Dr.Raghu Reddy, Mr.Madan Kumar Srinivasan. Seminar program conducted in LBRCE Aug 2013.

[16] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized Erasure Codes for Distributed Networked Storage," IEEE Transaction. Information Theory, vol. 52, no. 6 pp. 2809-2816, June 2006.

[17] Muneshwara M.S, Swetha M.S, Dr. Anil G.N ," A Smarter Way of Securing and Managing Data for Cloud Storage Applications Using High Throughput Compression in the Cloud Environment ", ISSN: 2327782 (Online) , International Journal of Advance Research in Computer Science and Management Studies. Volume 2, Issue 9, September 2014.

AUTHOR

Shivakumara T received B.Sc. degree from Gulbarga University and MCA from VTU in 2000 and 2007 respectively. During 2008 to present affiliated to BMS Institute of Technology and Management As Assistant professor at Department of MCA, actively involved in research area like Cyber security – detection and prevention. Several conferences and journal papers published. Three text books are in his credit. Pursuing the PhD in computer applications in VTU.

Muneshwara M S received B.E.and M.Tech from VTU in 2005 and 2012 respectively. During 2006 to present affiliated to BMS Institute of Technology and Management as Assistant professor at Department of CS&E, actively involved in research area like Distributed Network security and Cloud Computing. Several papers published in reputed Journals and conferences Pursuing the PhD in Computer Science under VTU.