International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Survey on: Fade Architecture for Secured Access and Assured Deletion in Cloud Storage

Nilam Musale[1], Vilas Gaikwad[2]

[1]Computer Engineering Department, JSPMNTC RSSOER, Savitribai Phule Pune University,

[2]Assistant Professor, Computer Engineering Department, JSPMNTC RSSOER, Savitribai Phule Pune University

Abstract: ***In order to reduce data management cost, we can outsource data backups off-site to third party cloud storage services. Third parties provide security guarantees for the outsourced data. FADE provides policy access control and assured deletion. Assured deletion aims to provide cloud client on option of reliably destroying their data backups upon request. 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. 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. A proof-of-concept prototype of FADE is implemented on atop Amazon S3, one of today's cloud storage services. Extensive empirical studies, and demonstration proves that FADE provides security protection for outsourced data, while introducing only minimal performance and monetary cost overhead.***

Keywords: access control, assured deletion, backup/recovery, cloud storage

1. Introduction

Cloud storage provides infinite storage space for clients to host data backups in a pay-as-you-go manner. The financial overhead of enterprises and government has greatly reduced because they can archive their data backups remotely to third party cloud storage providers instead of maintaining data centers on their own[1]. For example, SmugMug[2] is a photo sharing website. It has hosted terabytes of photos on Amazon S3 in 2006 and thus saved thousands of dollars on maintaining storage devices. Individuals can also archive their personal data to the cloud using various tools like Drop-box.

There are several security related issues when we store our sensitive data to third parties. In the proposed system, two security issues are addressed. First, access control should be provided to authorize parties only. Second, outsourced data should be permanently inaccessible to anybody (including data owner) upon request of deletion of data. The proposed system prohibits third party cloud service providers from mining any sensitive information of their client's data for their marketing purpose. The cloud storage providers have many backup copies of data for data loss reason. We cannot get assurance of whether cloud provides reliably remove all backup copies upon requests of deletion. The proposed system in this paper provides assured deletion of data upon request of deletion by data owner. In this paper author has presented FADE, A secure overlay cloud storage system that provides fine-grained access control and assured deletion for outsourced data on the cloud while working on cloud storage services.

The author has contributed following work.

1) A new policy based file assured deletion scheme is presented which surely deletes file when file access policies are revoked.
2) Two new features are implemented: fine grained access control based on attribute based encryption and fault tolerant key management with a quorum of key managers based on threshold secret sharing.
3) A working model of FADE is implemented on Amazon S3. Their implementation of FADE exports a set of API's that can be adapted into different data outsourcing applications.
4) The performance overhead of FADE is evaluated on Amazon S3.

Overall, the proposed system in this paper seeks to address the access control and assured deletion problems from a practical perspective.

2. Policy Based Assured Deletion

The FADE provides both access control and assured deletion for outsourced data. The design of FADE is based on the concept of policy-based file assured deletion. Time-based file assured deletion means that files can be securely deleted and are not allowed for access after a pre-defined duration. 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 which does 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 made 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. Time-based deletion to policy-based deletion is generalized as follows. Each file is associated 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.

3. FADE Overview

3.1 FADE Components

FADE is a system that provides guarantees of access control and assured deletion for outsourced data in cloud storage. FADE has following system components:

FADE clients. It is an interface that bridges the data source (e.g., file system) and the cloud. It performs encryption (decryption) to the outsourced data files uploaded to (or downloaded from) the cloud. It also interacts with the key managers to perform the necessary cryptographic key operations.

KeyManagers. FADE is built on a quorum of key Managers. Key manager is an entity that maintains policy-based keys for access control and assured deletion.

3.2 FADE Deployment

A FADE client is deployed locally with its corresponding data source as a local driver or daemon. It is also possible to deploy the FADE client as a cloud storage proxy , so that it can interconnect multiple data sources. In proxy deployment, one can use standard TLS/SSL for protection of communication between each data source and the proxy.

3.3 Cryptographic Keys

FADE has three types of cryptographic keys for protection of data files stored on the cloud:

Data Key: A FADE client creates and maintains data key is a random secret. It is used for encrypting or decrypting data files.

Control Key: A control key is related with a particular policy. A public-private key pair is used for representing it and the private control key is maintained by the quorum of key managers. Its purpose is to encrypt/decrypt the data keys of the files protected with the same policy.

Access Key: It is also associated with a particular policy, and is represented by a public-private key pair. It forms the basis of policy-based access control.

3.4 Security Goals

The security goals that FADE provides to protect the outsourced data files are as follows:

- **Policy-based access control**
- **Policy-based assured deletion**

4. FADE Design

Several cryptographic key operations that enable FADE to achieve our security goals are as follows:

Basic Operations of FADE

4.1 File Upload/Download

Figure 1 shows the file upload operation. The client first requests the public control key (ni, ei) of policy Pi from the key manager, and caches (ni, ei) for subsequent uses if the same policy Pi is associated with other files. Then the client generates two random keys K and Si, and sends {K}Si , Seii , and {F}K tothe cloud. Then the client must discard K and Si. To protect the integrity of a file, the client computes an HMAC signature on every encrypted file and stores the HMAC signature together with the encrypted file in the cloud. We assume that the client has a long-term private secret value for the HMAC computation.

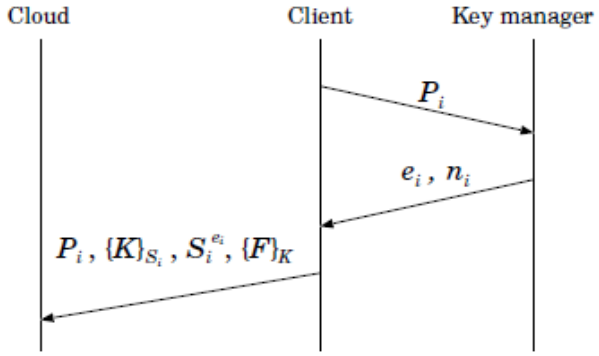

Figure 1: File Upload

Figure 2 shows the file download operation. The client fetches {K}Si , Sei i , and {F}K from the cloud. The client will first check whether the HMAC signature is valid before decrypting the file. Then the client generates a secret random number R, computes Rei , and sends Sei i ·Rei = (SiR)ei to the key manager to request for 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.

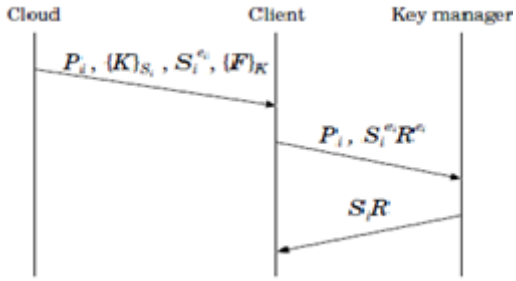

Figure 2: File Download

4.2 Policy Revocation for File Assured Deletion

If a policy Pi is revoked, then the key manager completely removes the private control key di and the secret prime numbers pi and qi. Thus, we cannot recover Si from Seii ,

and hence cannot recover K and file F. We say that file F, which is tied to policy Pi, is assuredly deleted. Note that the policy revocation operations do not involve interactions with the cloud.

4.3 Multiple Policies

FADE supports a Boolean combination of multiple policies. Two kinds of logical connectives: (i) the conjunction (AND), which means the data is accessible only when every policy is satisfied; and (ii) the disjunction (OR), which means if any policy is satisfied, then the data is accessible.

- **Conjunctive Policies.** Suppose that F is associated with conjunctive policies P1 ^ P2 ^ · · · ^ Pm. To upload F to the cloud, the client first randomly generates a data key K, and different secret keys S1, S2, . . . , Sm. It then sends the following to the cloud: {{K}S1}S2 · · ·Sm, Se11 , Se22 , . . ., Semm , and{F}K. On the other hand, to recover F, the client generates a random number R and sends (S1R)e1 ,(S2R)e2 , . . ., (SmR)em to the key manager, which then returns S1R, S2R, . . . , SmR. The client can then recover S1, S2, . . . , Sm, and hence K and F.
- **Disjunctive Policies.** Suppose that F is associated with disjunctive policies Pi1 _ Pi2 _ · _ Pim. To upload F to the cloud, the client will send the following: {K}S1 , {K}S2 , . . ., {K}Sm, Se11 , Se22 , . . ., Semm ,and {F}K. Therefore, the client needs to compute m different encrypted copies of K. On the other hand to recover F, we can use any one of the policies to decrypt the file, as in the above operations.

4.4 Policy Renewal

Policy renewal means to associate a file with a new policy (or combination of policies). In FADE, policy renewal merely operates on keys, *without retrieving the encrypted file from the cloud*. The procedures can be summarized as follows: (i) download all encrypted keys (including the data key for the file and the set of control keys for the associated Boolean combination of policies) from the cloud, (ii) send them to the key manager for decryption, (iii) recover the data key, (iv) re-encrypt the data key with the control keys of the new Boolean combination of policies, and finally (v) send the newly encrypted keys back to the cloud.

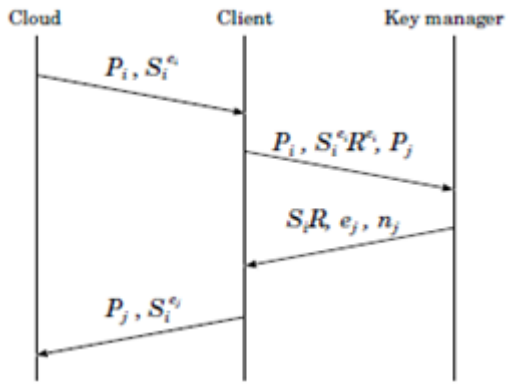

Figure 3: A special case of policy renewal - when policy Pi is renewed to policy Pj .

5. Implementation

A working prototype of FADE can be implemented using C++ on Linux.

5.1 Representation of Metadata

For each data file protected by FADE, we include the metadata that describes the policies associated with the file as well as a set of cryptographic keys.

5.2 Client

Client implementation uses four function calls to enable end users to interact with the cloud:

- Upload(file, policy)
- Download(file)
- Revoke(policy)
- Renew(file, new, policy)

5.2 Key Managers

A quorum of key managers, each of which supports two major types of functions:

(i) **Policy management**, in which a key manager creates or revokes policies, as well as their associated control keys (for assured deletion) and access keys (for access control)

(ii**) Key management**, in which a key manager performs the encryption or decryption on the (blinded) data key.

6. Evaluation

6.1 Time Performance of FADE

To identify the time overhead of FADE, running time of each measurement is divided into three components:

1. File transmission time
2. Metadata transmission time
3. Cryptographic operation time.

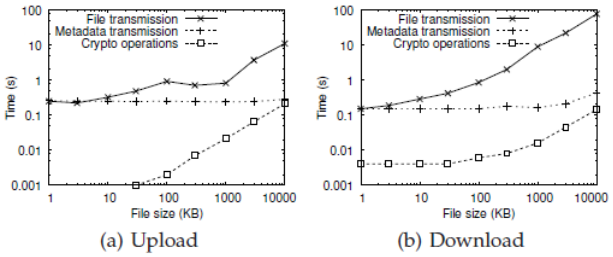

Figure 4: Performance of file upload/download operations.

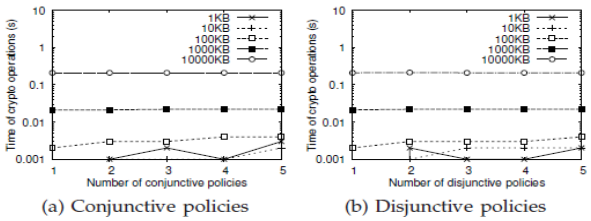

Figure 5: Performance of multiple policies

6.2 Space Utilization of FADE

Num. of policies	Num. of key managers				
	1	2	3	4	5
1	149	277	405	533	661
2	282	538	794	1050	1306
3	415	799	1183	1567	1951
4	548	1060	1572	2084	2596
5	681	1321	1961	2601	3241

Figure 6: Size of the policy metadata for conjunctive policies (in bytes)

Num. of policies	Num. of key managers				
	1	2	3	4	5
1	149	277	405	533	661
2	298	554	810	1066	1322
3	447	831	1215	1599	1983
4	596	1108	1620	2132	2644
5	745	1385	2025	2665	3305

Figure 7: Size of the policy metadata for disjunctive policies (in bytes).

7. Conclusion

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 is presented. Files are associated with file access policies that control how files can be accessed. Then policy-based file assured deletion is described, in which files are assuredly deleted and made unrecoverable by anyone when their associated file access policies are revoked. The essential operations on cryptographic keys so as to achieve access control and assured deletion are described. FADE also uses existing cryptographic techniques, including attribute-based encryption (ABE) and a quorum of key managers based on threshold secret sharing. A prototype of FADE is implemented to demonstrate its practicality. Its empirical study is done to measure its performance overhead when it works with Amazon S3. Experimental results show insights into the performance-security trade-off when FADE is deployed in practice.

References

[1] Yang Tang, Patrick P. C. Lee, "Secure Overlay Cloud Storage with Access Control And Assured Deletion," IEEE transactions and dependable and secure computing. Vol.9 No. 6 2012
[2] Y. Tang, P.P.C. Lee, "FADE: Secure Overlay Clod Storage with File Assured Deletion." In Proc. Of ICST SecureComm,2010
[3] J. Bethencourt, A. Sahai and B. Waters,"Cipher text-Policy Attribute Based Encryption," In proc. Of Symp. On Security and Privacy, May 2006
[4] R. Geambasu,T. Kohno, A. Levy, H. M. Levy."Vanish: Increasing Data Privacy with Self-Destructing Data ",In Proc. Of USENIX Security Symp.., Aug 2009.
[5] G. Ateniese, R. D. Pietro, L. V. Mancini, "Scalable And Efficient Provable Data Possesion," In. Proc. Of SecureComm, 2008

Author Profile

Ms. Nilam Musale received the B.E. degrees in Computer Science And Engineering from Dattajirao Kadam Textile Institute Of Engineering And Technology,Ichalkaranji in 2011. She is now pursuing ME in Computer Engineering from JSPM Narhe Technical Campus,Pune.