International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

A Novel Policy based Technology for Secure Cloud Computing

M.Likitha
likithalikim23@gmail.com
Dr.K.Ramakrishna
krkrishna.cse@gmail.com
Ch.Srilatha
srilatha.chinni10@gmail.com
Department of Software Engineering, Information Technology Sridevi Women's Engineering College

Abstract:

When we outsource data backup to third-partycloud storage services so as to reduce data management costs,security concerns arise in terms of ensuring the privacy andintegrity of out-sourced data. We tried to solve this issue bydesigning FADE (File Assured Deletion), a practical,implementable and readily deployable cloud storage systemthat focuses on protecting deleted data with policy-based fileassured deletion.FADE is built upon standard cryptographic techniques, suchthat it encrypts outsourced data files to guarantee theirprivacy and integrity, and most importantly, assuredly deletesfiles to make them unrecoverable to anyone (including thosewho manage the cloud storage)upon revocations of file accesspolicies. In particular, the design of FADE is geared towardthe objective that it acts as an overlay system that worksseamlessly atop today's cloud storage services.To demonstrate this objective, we implement a workingprototype of FADE atop Amazon S3, one of today's cloudstorage services, and empirically show that FADE providespolicy-based file assured deletion with a minimal trade-ofperformance overhead. Our work provides sin sights of how toincorporate value-added security features into current dataoutsourcing applications.

Keywords: Policy-based file assured deletion; cloud storage; prototype implementation

1. Introduction

Web based outsourced computing is getting popularityday by day because of its increasing users and servicedemands. It provides the various application and othercomputing capabilities as a service to the end user. Someof the examples of web based computing are cloudcomputing, Web 2.0, Mashups etc. These technologiesare core of application development which involves theintegration of various newly developed computingparadigms. Here the aim is towards making the software things available to users with lesser loads of managingthose applications and data.For using these services, the provider and users mustagree on some defined conditions known as service levelagreements (SLA's) [1]. For using the services someplatform is required which could be the browser forhighly demanded services and fewer configurations.Heavy application could be processed and suppliedeffectively using browsers and web based media.The computing could be one of the most demandedservices on the internet and hence requires the management of instances for individual users. Eachservice usages must be isolated from the other in terms oftheir

service usage policies and the kind of setting theyare demanded. It is a kind of intellectual collaboratedevolution for organizations by which the load ofmanaging server based technologies and capital cost arereduced and focus on their core business operations canbe increased. Such effective services are based on layeredand multi-tenancy models with dynamic scalability andvirtualized environment. So many firms are collaboratingthemselves for better service delivery and mitigating theassociated risk of data privacy and security.

Only acertain change in the behavior of this system causesoverall degradation in security controls and attackers orfabricates gets insight into the system. Even if a small view of data or its instances are mistakenly leavedunprotected after usages, it could be used for attacking ordistorting the normal working. Thus the probability ofoccurring attacks or data modification through someunauthenticated entity is very high.One should always focuses on the lifecycle of the datameans when the overall of living period of data is overthen it needs to be removed completely with all its localand permanent copies. Most of the organizations have theseveral policies for this data destructions based on thefixed time interval. But as of now the copies or replicasof data is getting multiplicatively increased so deletingall in a single go is very difficult. Also the deletion is notcomplete and some residues metadata remains at thelocation of the files from which the recreation of data canbe performed.Data destruction is the process of deleting the data andits overall components and copies when the lifecycle ofits operations is finished [2]. The deletion should be insuch a way that its reconstruction cannot be performedbut many organizations are unable to achieve suchbehavior and hence left a vacant space for attackers toregenerates the copies of original and forged some otherservices by the same.

2. Related Works

2.1Secure self-destruction scheme

A well-known method for addressing this problemis secure deletion of sensitive data after expirationwhen the data was used [3]. Recently, Caching et al.employed a policy graph to describe the relationshipbetween attributes and the protection class and proposed a policy-based secure data deletion scheme[4]. Reardon et al. leveraged the graph theory, B-tree structure and key wrapping and proposed anovel approach to the design and analysis of securedeletion for persistent storage devices [5]. Becauseof the properties of physical storage media, the above-mentioned methods are not suitable for the cloud computing environment as thedeleted datacan be recovered easily in the cloud servers [6].

2.2 Time-specific encryption

The time-specific encryption scheme TSE, proposedby Peterson et al. [7], was introduced as anextension of TRE [8]. In TRE, a protected datacan be encrypted in such a way that it cannot bedecrypted (even by a legitimate receiver who ownsthe decryption key for the ciphertext) until the time(called the release-time) that was specified by theencrypt or. Most of the previous TRE schemes thatadopt a time-sever model are in fact public-key TREschemes. They do not consider the sensitive dataprivacy after expiration [9][10][11]. In the TSE scheme, a time severs broadcasts a time instant key (TIK), a data owner encrypts a message into a ciphertext during a time interval, and a receiver can decrypt the ciphertext if the TIK is valid in that interval. Kasamatsu designed an

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

efficient TSE scheme by using forward-secure encryption (FSE) in which the size of the ciphertext is greatly small than that generated by the previous schemes [12]. The time interval may be considered as the authorization period of the protected data, and TSE schemes are able to meet this requirement. However, it is a tricky problem when the traditional TSE is used in the cloud computing environment: cloud computing environment needs a fine-grained access control [13], which cannot be provided by the traditional TSE schemes. How to achieve the time-specified ciphertext into a fine-grained access control level is a problem to be explored.

2.3 Policy-based Deletion:

We associate each file with a singleatomic file access policy (or policy for short), or moregenerally, a Boolean combination of atomic policies. Each(atomic) policy is associated with a control key, and all thecontrol keys are maintained by the key manager. Similar totime-based deletion, the file content is encrypted with a datakey, and the data key is further encrypted with the controlkeys corresponding to the policy combination. When apolicy is revoked, the corresponding control key will beremoved from the key manager. Thus, when the policycombination associated with a file is revoked and no longerholds the data key and hence the encrypted content of thefile cannot be recovered with the control keys of thepolicies. In this case, we say the file is deleted. The mainidea of policy-based deletion is to delete files that areassociated with revoked policies. is associated with acontrol key, and all the control keys are maintained by thekey manager.We review other related work on protecting outsourced datastorage.

3. Problem Statement

The shared data in cloud servers, however, usuallycontains users' sensitive information (e.g., personalprofile, financial data, health records, etc.) and needsto be well protected [17]. As the ownership of thedata is separated from the administration of them[14], the cloud servers may migrate users' data toother cloud servers in outsourcing or share themin cloud searching [15]. Therefore, it becomes a bigchallenge to protect the privacy of those shareddata in cloud, especially in cross-cloud and big dataenvironment [16]. In order to meet this challenge,it is necessary to design a comprehensive solutionto support user-defined authorization period andto provide fine-grained access control during thisperiod. The shared data should be self-destroyedafter the user-defined expiration time.

One of the methods to alleviate the problems isto store data as a common encrypted form. Thedisadvantage of encrypting data is that the usercannot share his/her encrypted data at a fine-grainedlevel. When a data owner wants to share someonehis/her information, the owner must know exactlythe one he/she wants to share with [6].We present policy-based file assured deletion, the majordesign building block of our FADE architecture. Our mainfocus is to deal with the cryptographic key operations thatenable file assured deletion. We first review time-based fileassured deletion. We then explain how it can be extended topolicy-based file assured deletion.

4. System Design

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

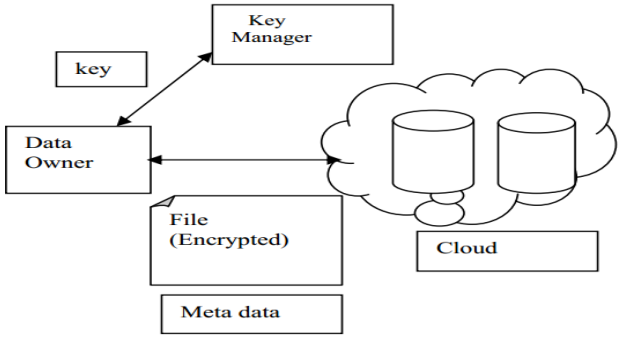

Fig. 1 FADE Architecture

Secure Overlay Cloud Storage with File Assured Deletionis a user-interactive app on the java platform. The project isits own entity and is derived based on the java.We design FADE, a practical implementable, and readilydeployable cloud storage system that focuses on protectingdeleted data with policy-based file assured deletion. FADEis built upon standard cryptographic techniques,such that itencrypts outsourced data files to guarantee their privacy andintegrity ,and most importantly, assuredly deletes files tomake them unrecoverable to anyone (including those whomanage the cloud storage)upon revocations of file accesspolicies.The application uses many different software interfaces.The Windows XP operating system is essential for theproduct to operate. Net Beans will be needed for thedevelopment portion of the project, and it will be utilizingthe java software packages. Communication is not neededby this application because it is based locally.The physical characteristics of the application consist ofvarious java app that run the java environmentthere is no need of communication hardware in ProjectFADE: Secure Overlay Cloud Storage with File AssuredDeletion.We define the metadata of FADE attached to individualfiles and also describe how we implement the data ownerand the key manager, and how the data owner interacts withthe storage cloud.

Representation of Metadata: For each file protected by FADE, we include the metadatathat describes the policies associated with the file as well asa set of encrypted keys. In FADE, there are two types ofmetadata:

File metadata: The file metadata mainly contains twopieces of information file size and hash. We hash theencrypted file and it is attached at the beginning. Boththe file metadata and the encrypted data file will thenbe treated as a single file to be uploaded to the storagecloud.

Policy metadata: The policy metadata includes thespecification of the Boolean combination of policiesand the corresponding encrypted cryptographic keys.Here, we assume that each single policy is specified bya unique 4-byte integer identifier. To represent aBoolean combination of policies, we express it indisjunctive canonical form, i.e., the disjunction (OR) ofconjunctive policies, and use the characters '*' and '+'to denote the AND and OR operators. Then we uploadthe policy metadata as a separate file to the storagecloud. This enables us to renew policies directly on thepolicy metadata without retrieving the entire file fromthe storage cloud. In our implementation, individualfiles have their own policy metadata, although weallow multiple files to be associated with the samepolicy. In other words, for two data files that are underthe same policy, they will have different policymetadata files that specify different data keys, and thedata keys are protected by the control key of the samepolicy.

5. Implementation

Our design is based on blinded RSA in which the dataowner requests the key manager to decrypt a blindedversion of the encrypted data key. If the associated policy issatisfied, then the key manager

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

will decrypt and return theblinded version of the original data key. The data owner canthen recover the data key. In this way, the actual content ofthe data key remains confidential to the key manager aswell as to any attacker that sniffs the communicationbetween the data owner and the key manager. For eachpolicy i, the key manager generates two secret large RSAprime numbers pi and qi and computes the product ni =piqi . The key manager then randomly chooses the RSApublic-private control key pair (ei, di). The parameters (ni,ei) will be publicized, while di is securely stored in the keymanager. On the other hand, when the data owner encryptsa file F, it randomly generates a data key K, and a secretkey Si that corresponds to policy Pi. We let {m}k denote amessage m encrypted with key k using asymmetric-keyencryption (e.g., RSA). We let R be the blinded componentwhen we use blinded RSA for the exchanges ofcryptographic keys. Suppose that F is associated withpolicy Pi. Our goal here is to ensure that K, and hence F,are accessible only when policy Pi is satisfied.

File upload: We set the private key value and also theexpiry date for the file and choose the file to be uploadedand then uploading is completed and we get a randomgenerated public control key. The file that is uploaded is inencrypted format by the RSA public key and is stored in thefolder in the cloud with the user folder name who isuploading.

File download: When we wish to download a file from thedata files we uploaded we need to first generate the controlkey for it that is done by entering the privacy key value weset along with the public control key that got generatedwhen file was uploaded this is result in giving us the publicand private keys by the RSA algorithm. The file can bedownloaded by entering the private control key that gotgenerated. The downloaded file gets stored in the folderdownloads in the cloud in decrypted format.

Request: The data owner can request the admin to give thedetails about the keys if incase he forgets them. The keysare sent to the mail id of the data owner in this way the keysare secure.

Delete File: The data owner tells the key manager topermanently revoke the specified policy. All filesassociated with the policy will be assuredly deleted byentering the password.

Policy Revocation for File Assured Deletion*:* If a policy Piis revoked, then the key manager completely removes theprivate key di and the secret prime numbers pi and qi. Thus,we cannot recover Si from Siei , and hence cannot recoverK and the file F . We say that the file F , which is tied topolicy Pi, is assuredly deleted. Note that the policyrevocation operations do not involve interactions with thestorage cloud.

Multiple Policies: In addition to one policy per file, FADEsupports a Boolean combination of multiple policies. Wemainly focus on two kinds of logical connectives: (i) theconjunction (AND), which means the data is accessibleonly when every policy is satisfied; and (ii) the disjunction(OR), which means if any policy is satisfied, then the datais accessible.

Policy Renewal: We conclude this section with thediscussion of policy renewal. Policy renewal means toassociate a file with a new policy (or combination ofpolicies). For example, if a user wants to extend theexpiration time of a file, then the user can update the oldpolicy that specifies an earlier expiration time to the newpolicy that specifies a later expiration time.

6. Experimental Results

International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

The experimental setup includes the usage of the opensource cloud service provider such as Jelastic cloud platform. The cloud platform allows the user to use thebasic functionalities and software's like TOMCAT, JAVAcompilers. These services will be provided free for over aperiod of 15 days. This minimum support is enough todeploy application on cloud. The database provided on thecloud will help to store the files uploaded and the encryptkeys used for each file, user is not aware of where the data isstored on the cloud, a separate private cloud environment iscreated and used for the experimental purpose. The figure 2shows the time taken for system to upload the file withdifferent sizes and figure 3 shows the time taken for systemto encrypt the file with different sizes

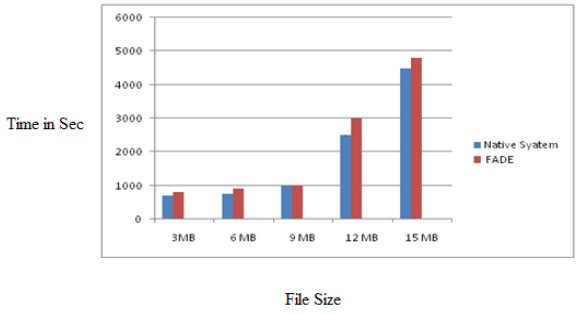

Figure 2: Comparison of time taken to upload a file with different sizes

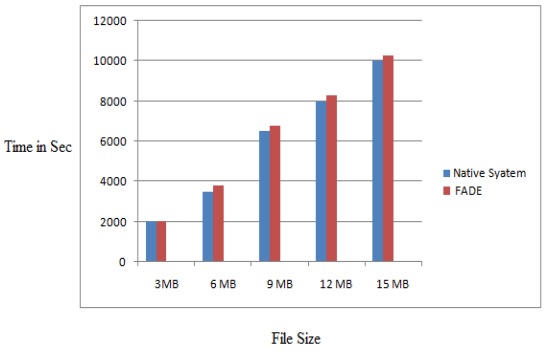

Figure 3: Comparison of time taken to encrypt a file withdifferent size.

7. Conclusion

We propose a cloud storage system calledFADE ,which aims to provide assured deletion for files thatare hosted by today 's cloud storage services .We presentthe design of policy-based file assured deletion ,in whichfiles are assuredly deleted and made unrecoverable byanyone when their associated file access policies arerevoked .We present the essential operations oncryptographic keys so as to achieve policy-based fileassured deletion .We implement a prototype of FADE todemonstrate its practicality, and empirically study itsperformance over head when it works with Amazon S3.Ourexperimental results provide insights into the performance security trade-off when FADE is deployed in practice.

8. References

[1]. Deyan Chen and Hong Zhao, "Data Security and PrivacyProtection Issues in Cloud Computing", in InternationConference on Computer Science and Electronics EngineeringIEEE Computer Society, DOI 10.1109/ICCSEE.2012.193, 2012.

[2]. Frank Simorjay, Ariel Siverstone and Aaron Weller, "ThMicrosoft approach to cloud transparency", awww.microsoft.com/twcnext, 2012.

[3] J. Reardon, D. Basin, and S. Capkun, "Sok: Secure datadeletion," in Proceedings of the 34th IEEE Symposium onSecurity and Privacy. IEEE, 2013, pp. 1–15.

[4] C. Cachin, K. Haralambiev, H.-C. Hsiao, and A. Sorniotti,"Policy-based secure deletion," in Proceedings of the ACMConference Computer and Communications Security. ACM,2013, pp. 152–167.

[5] J. Reardon, H. Ritzdorf, D. Basin, and S. Capkun, "Secure datadeletion from persistent media," in Proceedings of the 2013ACM Conference on Computer and Communications Security.ACM, 2013, pp. 271–284.

[6] J. Xiong, Z. Yao, J. Ma, F. Li, and X. Liu, "A secure selfdestruction scheme with ibe for the internet content privacy,"Chinese Journal of Computers, vol. 37, no. 1, pp. 139–150,2014.

[7] K. G. Paterson and E. A. Quaglia, "Time-specific encryption,"in Security and Cryptography for Networks. Springer, 2010,pp. 1–16.

[8] A. F. Chan and I. F. Blake, "Scalable, server-passive, useranonymous timed release cryptography," in Proceedings of theInternational Conference on Distributed Computing Systems.IEEE, 2005, pp. 504–513.

[9] J. H. Cheon, N. Hopper, Y. Kim, and I. Osipkov, "Provablysecure timed-release public key encryption," ACM Transactionson Information and System Security (TISSEC), vol. 11, no. 2,p. 4, 2008.

[10] A. W. Dent and Q. Tang, "Revisiting the security model fortimed-release encryption with pre-open capability," in Proceedings of the Information Security. Springer, 2007, pp. 158–174.

[11] R. Kikuchi, A. Fujioka, Y. Okamoto, and T. Saito, "Strongsecurity notions for timed-release public-key encryption revisited," in Proceedings of the Information Security and Cryptology.Springer, 2012, pp. 88–108.

[12] K. Kasamatsu, T. Matsuda, K. Emura, N. Attrapadung,G. Hanaoka, and H. Imai, "Time-specific encryption fromforward-secure encryption," in Security and Cryptography forNetworks. Springer, 2012, pp. 184–204.

[13] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing,"in Proceedings of the 29th IEEE International Conference on Computer Communications. IEEE, 2010, pp. 1–9.

[14] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, andP. S. Chen, "A full lifecycle privacy protectionscheme for sensitive data in cloud computing," Peerto-Peer Networking and Applications. [Online]. Available:http://dx.doi.org/10.1007/s12083-014-0295-x

[15] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research:A systematic review," Cloud Computing, IEEE Transactions on,vol. 1, no. 2, pp. 142–157, 2013.

[16] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, "Toward efficientand privacy-preserving computing in big data era," Network,IEEE, vol. 28, no. 4, pp. 46–50, 2014.

[17] J. Xiong, Z. Yao, J. Ma, X. Liu, Q. Li, and J. Ma, "Priam:Privacy preserving identity and access management scheme incloud," KSII Transactions on Internet and Information System(TIIS), vol. 8, no. 1, pp. 282–304, 2014.