Asian Journal of Applied Science and Technology (AJAST)
(Open Access Quarterly International Journal) Volume 2, Issue 2, Pages 777-782, April-June 2018

Self-Proxy Server Enhanced for Secure Mobile Data System

Abinaya.V[1] and Kiruthikadevi.K[2]

[1]PG Student, Department of Computer Science and Engineering, Nandha College of Technology, Tamilnadu, India.
[2]Assistant Professor, Department of Computer Science and Engineering, Nandha College of Technology, Tamilnadu, India.

Article Received: 01 March 2018 *Article Accepted: 09 April 2018* *Article Published: 28 April 2018*

ABSTRACT

The cloud database as a service is a new paradigm that can support many Internet-based applications Cloud Database is a service paradigm cause many research challenges in terms of security and cost evaluation from a tenant 's point of view, but its adoption requires the solution of information confidentiality problems. The new architecture for adaptive encryption of public cloud databases that offers a proxy free alternative to the system. The paper demonstrates the feasibility and performance of the proposed solution through a software prototype. Mobile cloud computing provides a new ecommerce mode for organizations without any investment. Cloud computing use distributed resource in open environment and it is important to provide secure keys to share the data for developing cloud computing applications. To ensure a correctness of user's data in the cloud, we propose an effective and secure distributed model including a Self-Proxy Server (SPS) with self-created algorithm. The model resolves a communication bottleneck due to re-encryption of a shared data in the cloud whenever users are revoked. It offers to reduce security risks and protect their resources because a distributed SPS dynamically interacts with Key Manager (KM) when the mobile users take on cloud services. To avoid this type of the situation the fuzzy authorization technique is used for the entered password partially matched, and the user can get the video that is the fake video. After that hacker stop retrying the username and the password. The original data is taken after real password is given by the real user.

1. INTRODUCTION

Managing and providing computational resources to client applications is one of the main challenges for the high-performance computing community framework. To monitoring resources existing solutions rely on a job abstraction for resource control, where users submit their applications as batch jobs to a resource management system responsible for job scheduling and resource allocation [1]. This usage model has served the requirements of a large number of users and the execution of numerous scientific applications. However, this usage model requires the user to know very well the environment on which the application will execute. In addition, users require administrative privilege over the resource to customize the execution environment job model. Manage and increasing availability of virtual machine technologies has enabled another form of resource control based on the abstraction of containers. A virtual machine can be leased and used as a container for deploying applications [2]. Under this scenario, users lease a number of virtual machines with the operating system of their choice; these virtual machines are further customized to provide the software stack required to execute user applications. This form of resource control is allowed the abstractions that enable many usage models, including that of batch job scheduling [3].

2. MAIN CONTRIBUTIONS

In the existing system, all data and metadata stored in the cloud database are encrypted and application running is a legitimate client can transparently issue SQL operations (e.g., SELECT, INSERT, UPDATE and DELETE) to the encrypted cloud database through the encrypted database interface. Data transferred between the user application and the encryption engine is not encrypted, whereas information is always encrypted before sending it to the cloud database. When an application issues a new SQL operation, the encrypted database interface contacts the encryption engine that retrieves the encrypted metadata and decrypts them with the master key. To improve performance, the plain metadata are cached locally by the client. After obtaining the metadata, the encryption engine is able to issue encrypted SQL statements to the cloud database, and then to decrypt the results. The results are returned to the user application through the encrypted database interface

1. Multi-user key distribution scheme is not proposed to provide data to the same group of users.
2. Encryption cost and thereby data transmission cost is more.
3. Same kind of encryption is maintained for all the data saved in the cloud nodes.

3. PROPOSED PROTOCOL

Like existing system, proposed system also manages the data using both cloud server side and client side. In addition, user group is maintained so that a single key is distributed to multiple users in the same group to reduce the key preparation overhead for each user.

This makes less computation overhead in both client and server side. Also, based on the security level, different data is encrypted with different encryption mechanism and allowed to secure the data in inexpensive manner.

1. Multi-user key distribution scheme is proposed to provide data to the same group of users.
2. Encryption cost and the data transmission cost is less.
3. Different kind of encryption is maintained for various data nodes.

4. ALGORITHM DESCRIPTION

4.1. SECURE MOBILE CLOUD COMPUTING WITH SELF-PROXY SERVER

A cloud is basically a large scale distributed system where a data owner's data is replicated over multiple servers for high availability. However, there are still a number of challenges because they are preventing the mobile users to take on cloud services. The model for a key distribution based on data re encryption is applied to a cloud computing system to address the demands of a mobile device environment.

While a user revoked, and he has decryption key he can access data still, thus to overcome from this problem here is a need of immediate re encryption of data by data owner. When re-encryption is done the newly generated, decryption keys are distributed to authorized users. A solution is to apply a distributed self proxy re encryption technique, so this scheme proposes Self Proxy Server (SPS). It coordinates and chooses keys by Key Manager (KM) whenever group membership changes.

The Key Manager (KM) generates and manages all data encryption, decryption and re encryption keys. It is provided live cloud computing by MCP and governed by Trusted Third Party (TTP). According to the self created algorithm, SPS uses re-encryption algorithm to transfer the encrypted format that can be decrypted by the user's private key. The user can download the encrypted data from the cloud and use the decryption key.

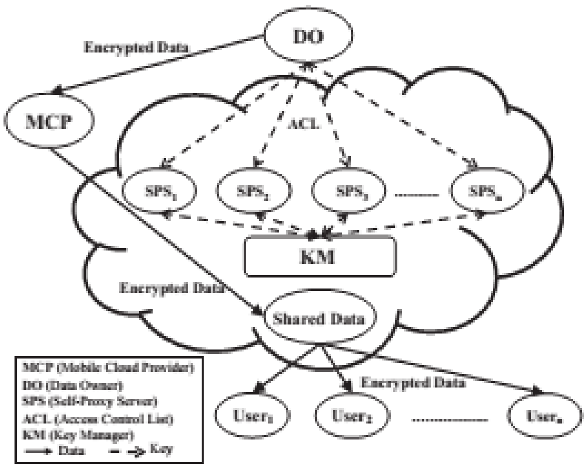

Fig 4.1 Architecture of Self Proxy Server

4.2 DISTRIBUTED CLOUD DESIGN USING KEY MANAGEMENT

To protect our data, we propose distributed SPS which interacts with KM whenever a mobile user requires cloud services. If the data in one of the cloud is corrupted, then the same data available in the other cloud could be accessed. Here the same data in different clouds would operate different keys from SPS which is under the control of KM. If same data sharing clouds work naturally, then users can choose SPS to re encrypt the data depending on the physical location. It can be applied to reduce the computation cost for the data owner. KM generates public Pu and private Su keys for each user belonging to the system and is Responsible for maintaining an ACL for enforcing the authorized user set. A data partition Pin the cloud is accessible by a user group Ug and belongs to the entire set of partitions. After confirming the access of user group Ug to data partition P, SPS interacts with KM. Even if they download it directly from the cloud, MCP and other users cannot decode the data with or without authentication. After all, read requests are initiated by users, this model is normally serviced through SPS which communicates with MCP. SPS receives it and communicates with KM whether the service for a mobile user is offered.

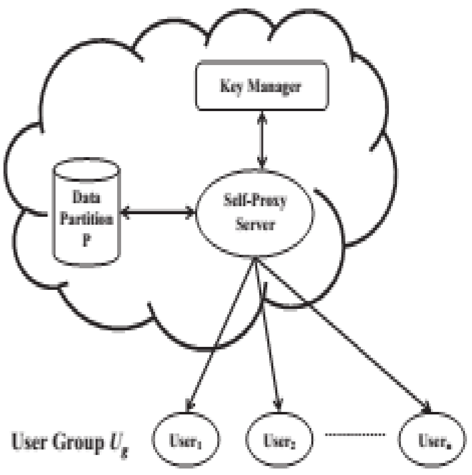

Fig 4.2 Distributed Cloud Design

4.3 FUZZY AUTHORIZATION SCHEME FOR CLOUD STORAGE

Data security is the one of the biggest concerns in adopting Cloud computing. In Cloud environment, users remotely store their data and relieve themselves from the hassle of local storage and maintenance. However, in this process, they lose control over their data. Existing approaches do not take all the facets into consideration viz. dynamic nature of Cloud, computation & communication overhead etc. In this study propose a Data Storage Security Model to achieve storage correctness incorporating Cloud's dynamic nature while maintaining low computation and communication cost. The only kind of guarantee is based on the level of trust between the cloud customer and the cloud provider and on the contractual regulations made between them such as SLAs, applicable laws and regulations of the involved jurisdictional domains. But even if the relation and agreements are perfectly respected by all participants, there still remains a residual risk of getting compromised by third parties.

5. SYSTEM ARCHITECTURE

5.1. Data owner Module

An entity who stores his/her data inside cloud storage and wishes to utilize cloud application services to process the data. A data owner must register with cloud storage provider and must be logged-in in order to upload, access data or authorize.

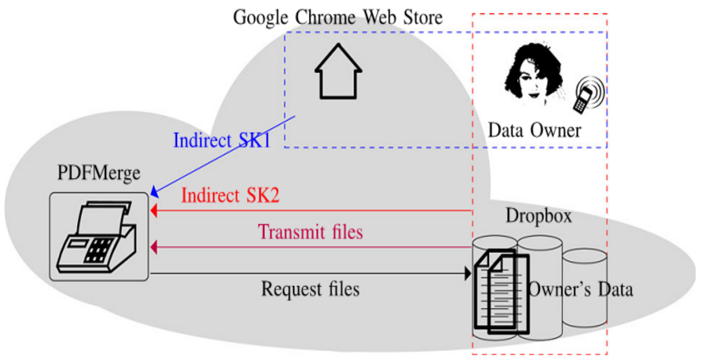

Fig 5.1 Fuzzy Authorization Scheme for Data owner, application service provider and Cloud Storage provider

5.2. Application Service Provider Module:

An entity to be authorized to access cloud storage data. It is an application software resides in vendor's system or cloud and can be accessed by users through a web browser or a special purpose client software. For example, PDF Merge is an online tool which can be used to merge several pdf files into one pdf file. With proper authorization, PDF Merge fetches the source pdf files from cloud storage. As a result, uploading files from data owner's local device is avoided.

5.3. Cloud Storage Provider Module:

An entity which supplies storage as a service to its clients and also provides access application programming interfaces to ASP when ASP holds a valid access token. Drop box and Just Cloud mentioned previously are examples of such entity.

5.4. Application Store (AS) Module:

An entity with which ASP must be registered to ensure itself's integrity and authenticity. Google Chrome Web Store is a typical application store. Data owner encrypts his data with a random symmetric key KE and encrypts KE with our modified CP-ABE scheme. Owner encapsulates ciphertext of KE and ciphertext of data as an archive and stores the archive in the CSP. Format of the archive is similarly defined. When owner needs to share data with ASP, he/she and CSP join together to issue ASP the indirect secret shares of file attributes while AS and owner collaborate to issue the indirect secret shares of application attributes. In this study, an indirect share contains a genuine secret share as its exponent or a part of its exponent.

6. CONCLUSION

In this experimental study, the existing system is describing the problem of secure authentication for storage in cloud. In this paper, proposed FA which carries out a flexible file-sharing scheme between an owner who stores the data in one cloud party and applications which are registered within another cloud party. The security analysis shows that our N-FA (Novel Fuzzy Authorized) scheme provides a through security of outsourced data, including confidentiality, integrity and secure access control. Novel-Fuzzy Authorized approach reduces the storage consumption compared to other similar possible authorization schemes. It also asserts that our scheme could efficiently achieve distance tolerance and realize fuzzy authorization. This work mainly addresses the reading authorization issue on cloud storage. And it results to enable the TPA to perform audits for multiple users simultaneously and efficiently.

REFERENCES

[1] Al Morsy M, Grundy J and Muller I, “An Analysis of The Cloud Computing Security Problem”, In Proceedings of APSEC 2010 Cloud Workshop, Sydney, Australia, November 2010.

[2] Armbrust M, Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, Patterson D A, Rabkin A, Stoica I, Zaharia M, Above the clouds: a Berkeley view of cloud computing, Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb. 2009.

[3] Buyya R, Ranjan R, Federated resource management in grid and cloud computing systems, Future Generation Computer Systems 26 (8) (2006) 1189–1191.

[4] Frank Gens, Robert P Mahowald and Richard L Villars. (2009, IDC Cloud Computing 2010.

[5] Kamara S and Lauter K, “Cryptographic cloud storage,” Financial Cryptography and Data Security, 2010.

[6] Khana N, Kiaha M, Khanb S and Madanic S, “Towards Secure Mobile Cloud Computing: A Survey”, Future Generation Computer Systems, vol.29, Issues 5, July 2013.

[7] Leavitt N, 2009, ‘Is Cloud Computing Really Ready for Prime Time?’ Computer, Vol. 42, pp. 15-20, 2009.

[8] Loknath S, Shivamurthy S, Bhaskar S and Shantgouda S, “Strong and Secure Re-Encryption Technique to Protect Data Access by Revoked Users in Cloud,” International Conference on Advances in Computer and Electrical Engineering (ICACEE'2012) Nov. 17-18, 2012 Manila (Philippines).

[9] Peter Mell, and Tim Grance, "The NIST Definition of Cloud Computing," 2009, http://www.wheresmyserver.co.nz/storage/media/faq-files/clouddef-v15.pdf, Accessed April 2010.

[10] Ramgovind S, Eloff M and Smith E, "The Management of Security in Cloud Computing", In PROC 2010 IEEE International Conference on Cloud Computing, 2010.