International Journal of Innovative Research in Engineering Science and Technology

APRIL 2018 ISSN 2320 –981X

DISTRIBUTED MOBILE CLOUD COMPUTING MODEL FOR SECURE MOBILE DATA CLOUD SYSTEMS

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

[1]Department of Computer Science and Engineering[1],

Nandha College of Technology[1],

Erode-638052, India.

abinaya101195@gmail.com[1]

[2]Department of Computer Science and Engineering[1],

Nandha College of Technology[1],

Erode-638052, India.

krithime@gmail.com[2]

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.

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].

Infrastructure base operating the local cluster can benefit from using Cloud providers to

improve the performance of its user's requests. Evaluate scheduling strategies suitable for a distributed cloud that is managed by proposed technology to improve its SQL operation with adaptive encryption data values. These strategies aim to utilize remote resources from the Cloud to augment the capacity of the SQL operation. The focus of paper is exploring the trade between performance improvement and cost using decryption and encryption key. As an application, they suggested private data banks: a user can store its data on an untrusted server in encrypted form, yet still allow the server to process, and respond to, the user 's data queries.

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

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

III. 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.

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

IV ALGORITHM DESCRIPTION

4.1 SECURE MOBILE CLOUD COMPUTING WITH SELF-PROXY SERVER

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). The data owner of MCP shares data to many other cloud users. The data is encrypted with a key from KM and then stored in the cloud along with Access Control List (ACL) indicating the user group. Upon access request from a user, the cloud communicates with SPS, based on ACL and SPS requests a self created algorithm. 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.

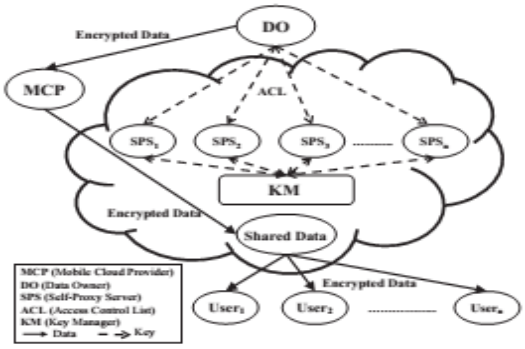

Fig 4.1 Architecture of Self Proxy Server

4.2 DISTRIBUTED CLOUD DESIGN USING KEY MANAGEMENT

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.

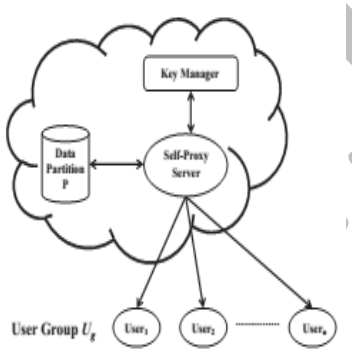

Fig 4.2 Distributed Cloud Design

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.

V 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.

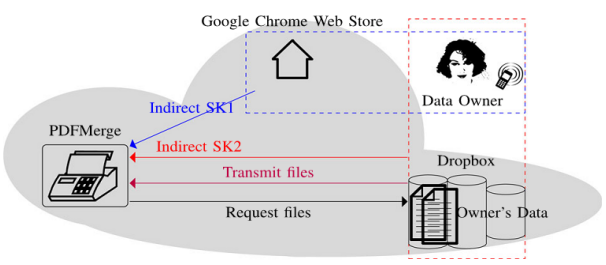

Fig 5.1 Fuzzy Authorization Scheme for Data owner, application service provider and Cloud Storage provider

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, PDFMerge is an online tool which can be used to merge several pdf files into one pdf file. With proper authorization, PDFMerge 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:

CSP (Cloud Service Provider) is trusted to provide storage services properly but may intend to access owner's data illegally. CSP may take advantage of the indirect shares that it possesses and query the other indirect shares so as to reconstruct the top secret. ASP (Application Service Provider) may try to decrypt the unauthorized files by utilizing the previous indirect shares issued to him. ASP is allowed to query for the indirect shares that he/she does not possess. AS (Application Store) which is involved in issuing the indirect application secret shares may try to access owner's data in the name of ASP. Since it knows about partial indirect shares of application attributes, he/she may query about the indirect shares of file attributes and try to obtain the complete indirect shares of application attributes.

VI 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.

LIST OF FIGURE

Figure1: Architecture of Self Proxy Server

Figure2: Distributed Cloud Design

Figure3: Fuzzy Authorization Scheme for Data owner, application service provider and Cloud Storage provider