International Journal of Master of Engineering Research and Technology, 2394-6172-(O)

Volume-5, Issue-2, February 2018

Enhanced the Security Strength of Data Retrieval in Cloud Computing Environments

Megha Saxena
Department of Computer Science & Engineering,
PCST Bhopal., (M.P.)
meghasaxena592@gmail.com

Prof. Parmalik Kumar
Department of Computer Science & Engineering,
PCST Bhopal., (M.P.)
parmalik83@gmail.com

Abstract
The storage and retrieval of data over cloud computing is big issue. For the storage and retrieval cloud computing used the concept of authentication and authorization. The process of authentication and authorization used primary and secondary authentication system. In primary authentication system used login id and password, in secondary login used the OTP and some other verification code. In both login systems trap the OTP code and primary information of user and the security process of cloud environment are compromised. In this paper proposed a model of secured access based on the concept of fake and genuine user. In the case of fake user, the received file is fake. This file is generated by the system. For the authentication of genuine and fake user used the concept of shared key concept. For maintain a data integrity and data security cloud computing adopt the process of third party auditor (TPA).

Keywords: - ***Cloud Computing, TPA, Public Auditing, Cyclic Shift Key, Key Generation Techniques***

INTRODUCTION

The cloud computing is the concept of delivery of computing as a service rather than product, the computer resources, software and information shared instead of other devices. In the idea of cloud computing the user of cloud outsources its data on to the cloud, and then the third-party auditor is going to check authorization of that user to access the cloud [3]. Data storage paradigm in "cloud" brings many challenging issues which have profound influence on the usability, reliability, scalability, security, and performance of the overall system. One of the biggest concerns with remote data storage is that of data integrity verification at un-trusted servers [1, 3]. The cloud storage has a lot of problems about the security and data Integrity. So, they need to prevent the all problems. In cloud storage users can remotely store their data and enjoy the on-demand high quality applications and services from shared resources, without the burden of local data storage and maintenance. Users are not able to check his data again and again from the cloud storage it is secure or not. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor to check the integrity of outsourced data and be worry-free[4,5,9].

In Cloud Computing, the remotely stored electronic data might not only be accessed but also updated by the clients, e.g., through block modification, deletion, insertion, etc. Unfortunately, the state of the art in the context of remote data storage mainly focus on static data files and the importance of this dynamic data updates has received limited attention [2, 6]. According to the role of the verifier in the model, all the schemes available fall into two categories: private verifiability and public verifiability. Achieving higher efficiency, schemes with private verifiability impose computational burden on clients. On the other hand, public verifiability alleviates clients from performing a lot of computation for ensuring the integrity of data storage. To be specific, clients are able to delegate a third party to perform the verification without devotion of their computation resources[11, 12]. To ensure cloud data storage security, it is critical to enable a TPA to evaluate the service quality from an objective and independent perspective. Public auditability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. Another major concern is how to construct verification protocols that can accommodate dynamic data files[8, 10]. The rest of paper describe as section II

Paper ID:- 2018/02/IJMERT/02/318

discuss the system model design. In section III describe the experimental analysis. in section IV describe the result analysis and finally discuss conclusion and future work.

II. MODEL DESIGN

There are three main party of our design model. (i) CSP (cloud service provider), who control the access and management of data control over the cloud. (ii) third party auditor(TPA) who gives the trust value of user and cloud server. UI (user interface) the user proceeds the request for the data retrieval and storage in participation of cloud server provider and TPA.

ENCRYPTION PROCESS: Performed at UI site or CSP site, they can perform the process of encryption for the generation of session key. The process of encryption done by the cyclic shift key generation technique. the cyclic shift key generation technique is emerging key generation technique by symmetric key technique.

VERIFYING DATA INTEGRITY: Simply downloading the data for integrity verification is not a practical solution due to expensiveness in I/O cost and unsafe files transfer across the network and may lead to new vulnerabilities [14]. Moreover, legal regulations, such as (HIPAA) [50], further demand the outsourced data not to be leaked to external parties (e.g. TPA). So applying encryption before outsourcing is the most preferred way to mitigate the privacy concern. Along with MD5 and MAC, Proof of storage [11] is widely used protocol for the purpose of checking integrity of data stored on remote server. The algorithms can be run any number of times as user wants, and they do not result into too much communication or computations overhead. It produces a very small amount of information (irrespective of the size of the data file) which can be exchanged between user and Cloud, any number of times.

1. MODEL DESCRIPTION

The overall operation of the entire model is divided among following main seven phases.

1. User Registration phase

2. Pre-storage phase

3. Storage phase

4. Grant access rights

5. Data download phase

6. Data verification phase

The below figure shows Security Model for Data Storage which contain three entities and the operation occurred between them. Then, let we discuss the phase mentioned above in detail.

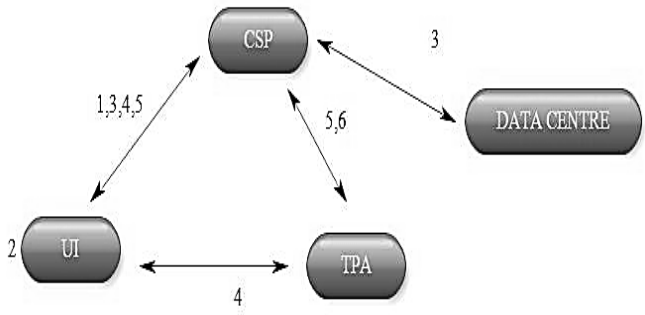

Figure 1: Security Model for public auditing over cloud

2. REGISTRATION PHASE

This phase is responsible to register a UI with the Cloud Service Provider. This task can be achieved by using following set of operations.

1.1 Request for registration (UI →CSP):

IDCSP || E(PRUI, IDUI || E(PUCSP, (Ni ||Tj || "Request=NewReg")))

1.2 Registration acknowledged (CSP → UI):

E(PUUI, (IDUI || Ni || Tj || "Response=Accepted/Rejected"))

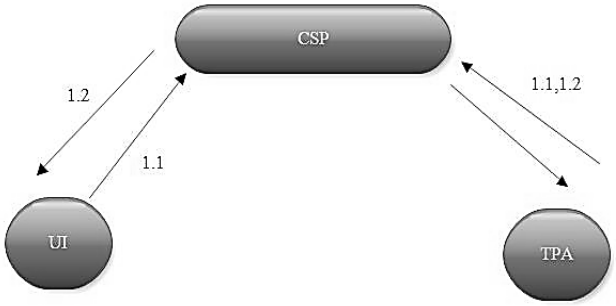

Figure 2: Registration Phase of user over cloud network.

During registration phase user send a registration request contain time stamp, ID of sender and this combined package is encrypted using an cyclic key of Sender. At the Cloud server, it first decrypts the file using sender's cyclic key which give assurance that the sender is authenticate person and then the server decrypt the packet using its own private key and get the request's data. After processing Cloud server send an acknowledge

back to the sender with status either accepted or rejected by encrypting packet with cyclic key of receiver which is decrypted By private key of receiver and user check its request status from acknowledge. Here whatever information send the client is stored for future verification the table structure is shown below.

Filed	Name
Field Details	
User ID of the user UI	ID
Uname User name	
Password Password of the user	

2.1 Encrypt the file (Done by UI):

E(KF,F)

2.2 Calculate the Cyclic code & encrypt it (Done locally by UI):

HORG=E(KF, H(E(KF,F)))

This phase responsible to produce a secure data which can place on to the CSP. First, User encrypt the file which is available on local premises using and symmetric key. The key is produced by any of encryption schemes available. Once file encryption done the user calculate a hash code for the encrypted file. This hash code is generated using an encoding algorithm (e.g. SHA256, MD-5) Available. Client store the calculated hash into database for the future verification.

3. DATA DYNAMICS PHASE

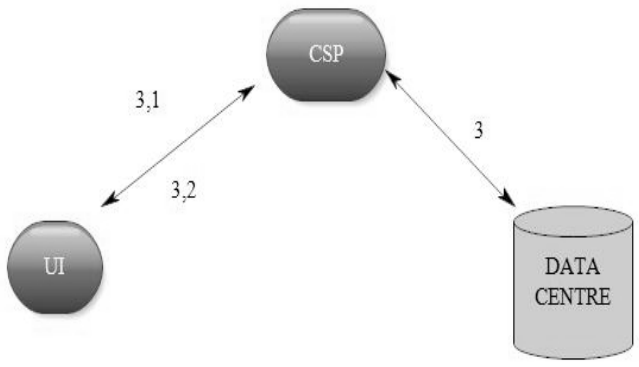

Figure 3: Data storage and data retrieval phase

The main responsibility of this phase is to store the generated file in second phase is on CSP securely. This task in implemented using following set of operations

3.1 Request to store (UI → CSP):

C= IDUI || E(PRUI, IDUI || E(PUCSP, E(KF,F) || Ni || Tj || HashType ||

equest=FileStore"))

3.2 Request to store confirmed (CSP → UI):

E(PUUI, (FileID || Receipt || Ni || Tk || "Response=Accepted/Rejected"))

During storage phase the client send "File Store" request to store the encrypted file with some more parameter like Sender's ID, Hash type Etc. by encrypting using cyclic key of CSP so only CSP can decrypt it and again encrypt same using private Now, At the CSP it decrypts the incoming coupon. First CSP decrypt coupon using cyclic key of sender and then again decrypt it using its own private key. It gets the data which it store into the database and sends the confirmation to the senders by encrypting it using cyclic key of sender which contain status of request with the receipt. CSP maintain note of the files which is stored by different user on server into the database.

4. GRANT ACCESS RIGHTS PHASE

This is a one of the important phase of Model. During the phase the requested rights by the UI's are May granted or denied y the data owners. This task is implemented using some set of operations given below.

4.1 Grant access rights (UI → UI):

IDUI||E(PRUI, IDUI||E(PUUI, (FileID||AR||EncrType||HashType|| KF)))

4.2Make CSP aware (UI → CSP):

IDUI||E(PRUI, IDUI||E(PUCSP, (FileID|| IDUI||AR)))

During Grant rights phase first Request sends by the requester on to the CSP which is stored by the server and intimate of the same to be given to Owner when owner makes check its pending rights. When owner finds a request for its file then either it grant or deny the grant request. In case of granting the request, owner sends File ID, Hash type, Encryption algorithm used Etc. by encrypting with cyclic key of requester so that particular user can only decrypt it. Again, owner encrypt the same

using a private key of owner which gives surety to the receiver about authorization of requester. At the last the owner makes also aware to CSP about this by sending file Id, the user id whose request are fulfilled and the rights assigned Etc. encrypting by the cyclic key of CSP, and again use its own private key for encryption for the purpose of authentication. On the receiving this from the owner the CSP makes the necessary change into the database.

5. DATA DOWNLOAD PHASE

This phase contains the fundamentals for downloading a file from the CSP by the UI's.

The UI can only able to download the file from CSP if owner granted the rights for its

Request. This is achieved using following set of operations.

5.1 Request for data (UI → CSP):

IDuser||E(PRuser, Iduser || E(PUCSP, (FileID || Ni || Tj || "DownloadReq")))

5.2 Data response (CSP → UI):

E(PUuser, E(KF, F) || Ni || Tj || "DownloadResponse=Accepted/Rejected"))

Once the user gets the permission from the owner the user can send the request to CSP for downloading of files. This request contains parameters like File ID, Time, ID of sender Etc. and whole request are encrypted using cyclic key of CSP, so only CSP can decrypt it. Again, the sender also encrypts same using its own private key which provides an authentication at the CSP side.

6. DATA VERIFICATION PHASE

This is the phase by which user can ensure about the correctness for him/his files. This phase contains following set of operations.

6.1 Data verification request (UI → CSP):

IDuser||E(PRuser, IDuser||E(PUCSP, (FileID|| Ni || Tj || "VerifyReq")))

6.2 CSP computes the hash code & encrypts it:

HCSP=E (PUuser, H (E(KF, F)))

6.3 Data verification response (CSP → UI):

E(PUuser, HCSP || HORG || Ni || Tj))

During this phase the sender sends a verification request contains parameter like ID of file which they want to verify with the sender's ID by encrypting using cyclic key of CSP, so only CSP can decrypt it. Again, this whole is encrypted using private key of sender so CSP can ensure that the sender is genuine. Once an CSP get the request it calculate the hash code for the requested file using an encoding scheme (Which is sent previously by owner) and encrypt it using a cyclic key of requester. This calculated reply is sent to the Requester where it is verified by the requester.

7. DATA UPDATE PHASE

This phase is responsible to update the existing file on to the CSP. This phase working in

somewhat same fundamental like in phase2.

7.1 Data update request (UI → CSP):

HORG=E(KF, H(E(KF,F)))

IDUI||E(PRUI, IDUI||E(PUCSP, E(KF, F) || HORG || Ni || Ti || FileID))

7.2 Data update confirmation (CSP → UI):

E(PUUI, (Ni || Tj || FileID))

Once user gets the file from the CSP, user may make the changes into the file. Now this updated file must be store back to CSP so, for this first the user encrypts the file using shared symmetric key and calculate Hash of the updated file. Now user sends the request for updating file with the parameter like File ID, User ID Etc. by encrypting it using CSP's cyclic key so only it can access. Again, sender encrypt it using a private key of sender so that CSP can authorize the sender and if sender is genuine then CSP update the database and makes mark about updation.

After updating CSP sends confirmation to the user contain File ID and time by encrypting it using user's cyclic key.

III EXPERIMENTAL ANALYSIS

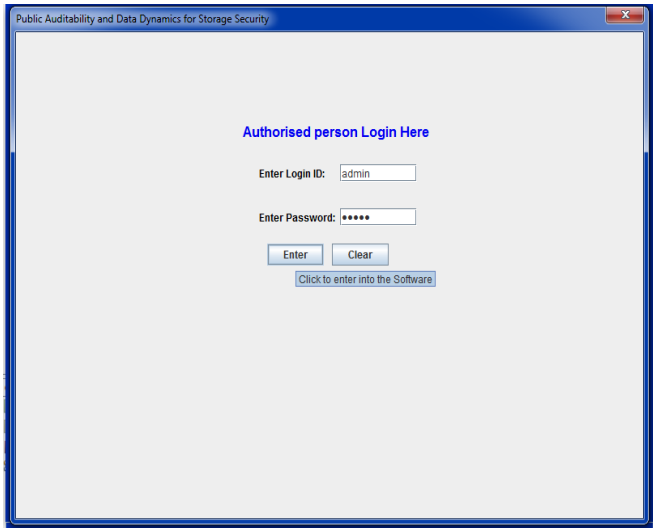

Figure 4: Shows that the authentication for user login.

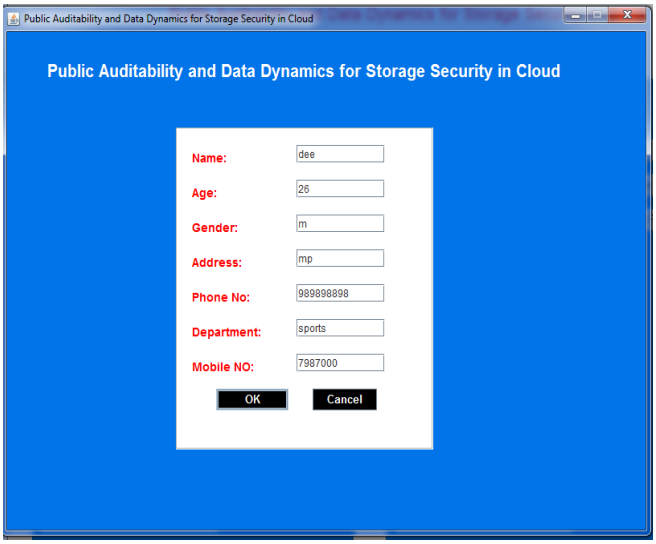

Figure 5: Shows that the completion new user window for public auditability in storage cloud system for large amount of database.

IV PERFORMANCE ANALYSIS

Types of File	File Name	Hit Ratio in %	Miss Ratio in %	Data Type value
Original File	Abc.txt	0.9	0.1	False
Fake file	Bca.txt	0.85	0.15	True

Table 5.1: Shows that the comparative performance for original and fake files based on number of hit and miss ratio in percentage value for the Abc and Bca file.

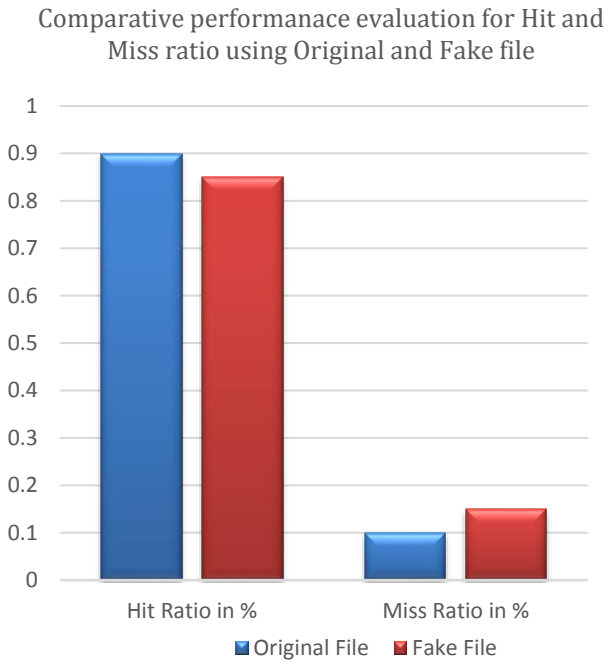

Figure 6: Shows that the comparative performance evaluation graph for original and fake files based on number of hit and miss ratio in percentage value for the Abc and Bca file.

DRDP Method		**RSA Based instantiation**		**Cyclic Based**	
Block Data size	Compu-tation Time	Block Data size	Compu-tation Time	Block Data size	Compu-tation Time
0	200	0	220	0	210
20	220	20	240	20	230
40	240	40	260	40	250
60	260	60	280	60	270
80	280	80	300	80	290
100	300	100	320	100	310
120	320	120	340	120	330
140	340	140	360	140	350
160	360	160	380	160	370

180	380	180	400	180	390

Table 2: Shows that the comparative performance for Computation time on the basis of block size using methods DRDP, RSA Based and Cyclic Based.

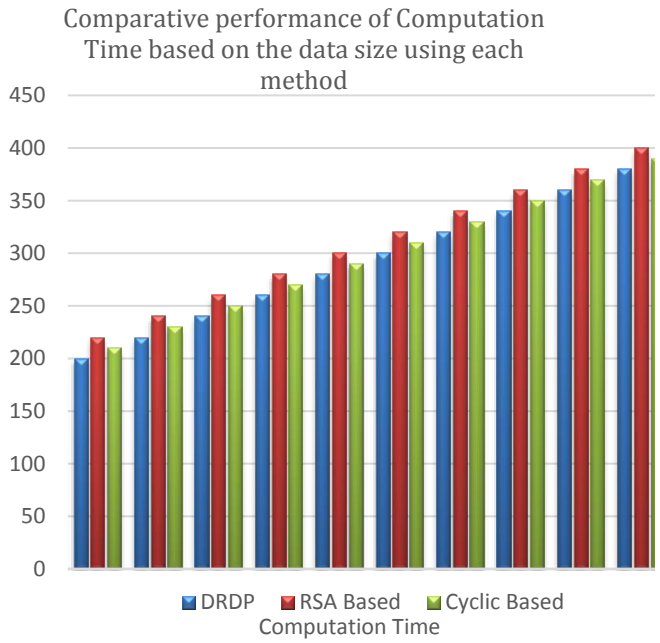

Figure 7: Shows that the comparative performance for Computation time on the basis of data block size using each method like DRDP, RSA Based and Cyclic Based, here we find the value of computation time for respectively block size and methods.

V CONCLUSION & FUTURE WORK

The cloud service provides is on duty to ensure the security of cloud data storage and to ensure maximum protection. Service providers have the responsibility to ensure the public data integrity and isolation protections are put in place to mitigate the risks users pose to one another in terms of data loss, misuse, or privacy violation within the cloud. Again, from the cloud service provider's perspective, there should be an active monitoring mechanism in place to allow for effective planning and implementation of services. This also serves as a means to respond to events quickly and more efficiently. Cloud users on the other hand must ask and be clear about their responsibility for their security. By this, it is recommended to SMEs to ask question about their security when engaging a service provider. Questions about; what information is stored on a system, where is the information stored, who can access the system, what they can access and appropriate access mechanism are good to clear any doubt about services providers. Identifying controls that address the lack of direct access to data and information is the foundation of SME's strategy for entering the cloud and allows the organization to consistently approach security needs based on the workloads and granular data represented in their cloud efforts.

References

[1] Jin Li, Xiao Tan, Xiaofeng Chen, Duncan S. Wong, Fatos Xhafa, "OPOR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices" IEEE 2015, Pp 195 205

[2] Qian Wang, Kui Ren, Member, Wenjing Lou, Jin "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" IEEE 2011 847 - 859

[3] Meera Chheda, Anmol Achhra, Priyanka Vaswani, Rajeshwari Agale, Vidya Bhise. "Public Auditing for The Shared Data In The Cloud". International Journal of Advance Foundation and Research in Computer (IJAFRC) 2015 Pp 724-728.

[4] Prof. N.L. Chourasiya, Dayanand Lature, Arun Kumavat, Vipul Kalaskar, Sanket Thaware. "Privacy-Preserving Public Auditing for Secure Cloud Storage" International Journal of Engineering Research and General Science, 2015 Pp 744 -748.

[5] R.Guruprasath, M.Arulprakash "Privacy Preserving Public Auditing For Shared Data With Large Groups In The Cloud" Journal of Recent Research in Engineering and Technology 2015 Pp 40-46

[6] Mrunali Pingale, Prof. Jyoti Pingalkar "Security Preserving Access Control Mechanism In Public Clouds Using PANDA Security Mechanism" iPGCON, 2015 Pp 1-5.

[7] Pradnya Chikhale, Namrata Dwivedi, Parna Dutta, Aparajita Sain, Vrunda Bhusari "Enhancing Data Storage Security In Cloud Computing Using PDDS Technique" PISER 2014 Pp 53-59.

[8] J.Aparna, Mr.R.Sathiyaraj "Auditing Mechanisms for Outsourced Cloud Storage" International Journal of Computer Science and Mobile Computing, 2014, Pp 219-229.

[9] Ch. Rajeshwari, S. Suresh "An Efficient PDP Scheme For Distributed Cloud Storage To Support Dynamic Scalability On Multiple Storage Servers" International Journal of Science Engineering and Advance Technology, 2014 Pp 985-988

[10] Betzy K. Thomas, M. Newlin Rajkumar "A Dynamic Public Auditing Security Scheme To Preserve Privacy In Cloud Storage" IJSHJE 2013, Pp 93-97.

[11] GuangyangYang, Hui Xia, Wenting Shen, Xiu xiu Jiang,Jia Yu "Public Data Auditing with Constrained Auditing Number for Cloud Storage" 2015 IJSIA Pp 21-32.

[12] Jian Yang, Haihang Wang, Jian Wang, Chengxiang Tan, Dingguo "Provable Data Possession of Resource-constrained Mobile Devices in Cloud Computing" Journal Of Networks, 2011 Pp1033-1040.