ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)

Implementation and Evaluation of Privacy Preserved Public Auditing in Cloud Storage System

[I]Reshma B, [II]Dr.Sanjay Srivastava
[I]PG Student, Dept. of CS&E, MGM CoET, Noida, Uttar Pradesh, India
[II]Associate Professor, Dept. of CS&E, MGM CoET, Noida, Uttar Pradesh, India

Abstract
Cloud computing is a set of IT services provided by a third party provider who owns the infrastructure to the customers in need of it over the network. In still simpler terms, cloud computing refers to the delivery of computing services such as software, storage, servers, networking, analytics, databases etc. Users can store their data on cloud and access it independent of location and time. Before sending the data to the cloud the data block would be signed to provide security to user's data. One of the primary aims of cloud computing is to ensure the user to use the cloud storage as if it is local, not worrying about data integrity. Thus arises the indispensability to introduce public audit ability with the help of a Third Party Auditor(TPA) to audit the stored data on behalf of the user. In the absence of the data owner, to repair the wrong server detected by the auditing process we also introduce a proxy.

Keywords
Cloud Storage, Public Auditing, Third Party Auditor, Regenerating Codes, Proxy.

1. Introduction

Cloud computing is the latest technology which enables convenient network access to a pool of configurable computing resources. Cloud Computing is the most awaited technology which would allow users to remotely store data in the clouds and to enjoy the on-demand high quality services and applications from a shared pool of configurable computing resources. Outsourcing their data on cloud relieves users from the burden of local data storage and its maintenance aspects. But, the fact that users would no longer have physical possession of the large size of outsourced data makes the data integrity protection in Cloud Computing a very challenging and potentially redoubtable task, specially for those users with limited computing resources and capabilities. Therefore, enabling public audit ability for data storage on cloud is of utmost necessity for users can opt for an external audit party to check the data integrity whenever necessary. Secure introduction of a Third Party Authenticator(TPA) involves the following basic requirements to be satisfied:

1) Efficient auditing of the data stored on cloud has to be done by TPA without demanding any local copy of the data. It is also necessary that TPA in no way introduce any additional on-line burden to the user of the data.
2) The implementation and use of Third Party Auditor should be in no way harmful to user's data privacy. In this paper, we employ and combine homomorphic authenticator which is public key based with a random masking to achieve privacy-preserving public cloud data auditing system, to meet all the above requirements. To regenerate the failed authenticators, we also introduce a proxy. Our system is privacy preserving since neither the proxy nor the TPA has access to the user's data stored on cloud. Security and performance analysis of our system proves that the proposed scheme is secure and highly efficient.

II. Implementation

The proposed scheme has three phases basically. They are as follows:

- Setup Phase
- Audit Phase
- Repair Phase

Setup Phase

The data owner generates a random signing key pair, two random elements and computes secret parameter and public parameter. The data owner sends encrypted random variable to the proxy using proxy's public key, then the proxy decrypts and stores it locally upon receiving. The data owner uniformly chooses a random identifier, a random symbol, one set with an element and a file tag. Recall that the original file is split into m blocks. The client computes and stores p$ coded blocks among p cloud servers. For simplicity, assuming each segment of block as a single symbol, our signature is generated simultaneously with the encoding procedure.

Audit Phase

TPA issues an audit message to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will create a response message by executing Genproof using F and its verification metadata as inputs. The TPA then verifies the response by cloud server via Verify Proof.

A owner is a person who can access resources from the cloud. The owner would first register to the interface to get the services with the valid username and password. In order to correctly audit the integrity of the entire data, a public verifier needs to choose the appropriate public key for each block. Then they can request for the file to the cloud service admin. There will be a third party auditor who performs the integrity checking of the data before providing it to the owner or the users. This is done by 1st splitting the data into blocks and then performing integrity check. The owner has the option of downloading the verified file and also uploads new files.

Repair Phase

The data owner delegates repair rights to a proxy and moves to offline as soon as file upload procedure is completed. Proxy takes charge of reparation process of faulty servers. When Third Party Auditor detects corruption in any server, an alert would be sent to proxy server which in turn would trigger a repair procedure. Most importantly, the l blocks downloaded for using for repairing would be checked for correctness prior to using for regeneration

© IJARET All Rights Reserved

to avoid pollution attack.
What follows is the Implementation of the proposed system explained in detail :

User registration

- *Register.jsp*

Register.jsp page displays the user registration and redirects to register1.jsp
The details of user like first name, last name, email, country, mobile no, user name and password are stored in the owner table.

User login

- *Ologin.jsp*

Displays the owner login page and then redirects to ologin1.jsp
Takes parameters such as username and password and then checks the validity in the owner table.
Once the owner is logged in, upload the text file where the owner is provided with two options - view file and send to auditor

File upload

- *File upload1.jsp*

Takes the parameters such as file name, date. Fkey is generated and stored in file11 table.

- *File upload2.jsp*

The file uploaded is divided into blocks and stored in tables like d1, d2 and d3. Redirecting to upload-success.jsp shows file upload success page.

View file

- *View_ownerfile.jsp*

Takes the file name parameter, on submission it gets redirected to view_ownerfile1.jsp

- *View_ownerfile1.jsp*

Query the request with file name in tables d1, d2 and d3 then redirects to file_view1.jsp

- *File_view.jsp*

Select the fname and date which would be in encrypted file format and select decrypt file and key from table file11.

- *Status_update.jsp*

Update file11 set and successfully send the permission to decrypt the file.

- *Vv.jsp*

Store the file block wise and store in d1, d2 and d3 then redirects to mergefile.jsp

- *Mergefile.jsp*

Shows the options- merge and download then redirects to check.jsp which takes fkey for merging the file, check for fkey in file11 table. If its valid, redirects to view_mergefile.jsp where one can view and download owner's original file.

- *Download.jsp*

Select decrypt file and key from table file, responds to download1.jsp which takes file name and then a valid user can download file.

Auditing process

- *Auditor_login.jsp*

Displays the auditor login page.

- *Auditor_logincheck.jsp*

Takes the parameters, user name and password from auditor then either redirects to auditor_page.jsp else displays the message- enter correct user name and password.

- *Auditor_page.jsp*

Displays the page with options like view auditing file, file issues and file issues cleared.

- *Auditing_files.jsp*

Display the file details like id, file name, date and file size, stored in table file11.

- *Issues_page.jsp*

Displays file details with the options- issues send to owner which inturn redirects to emailsend.jsp.

- *Issues_clearedpage.jsp*

Displays the file details which are cleared and send owner directs to emailsend1jsp.

- *Emailsend.jsp*

Will create a dummy mail id which takes recipient email id and send message- unfortunately your file is deleted from the cloud we will recover soon.
This display message would be either successfully sent otherwise may get failed.

Cloud process

- *Cloud_logincheck.jsp*

Take the parameters such as username and password, then, redirects to cloud_page.jsp

- *cloud_page.jsp*

Displays the page with options server1 file, server2 file and server3 file.

- *server1.jsp*

The file detail is displayed with file id, file name, file date, and file size along with a delete option. Once the file is deleted this directs to delete_server1jsp which updates d1 table set count.

- *Server2.jsp*

The file detail is displayed with delete option. If the file is deleted, then delete_server2.jsp updates d2 table set count.

- *Server3.jsp*

The file detail with delete option is displayed. once If the file is deleted, then delete_server3.jsp d3 table is set count.

Proxy server

- *Proxy.jsp*

Display proxy login page.

- *Proxy_logincheck.jsp*

Take the parameters username and password and redirects to proxy_page.jsp

© IJARET All Rights Reserved

ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)

- *Proxy_page.jsp*

Displays the proxy page with options- proxy view server1 file, proxy view server2 file and proxy view server3 file.

- *Proxy_view_server1.jsp*

Display server1 file details with send to auditor and regenerate code options.

- *Proxy_view_server2.jsp*

Display server2 file details with send to auditor and regenerate code options which would in turn redirect to regenerate_server2.jsp

- *Proxy_view_server3.jsp*

Display server3 file details with send to auditor and regenerate code options which redirects to regenerate_server3.jsp

- *Regenerate.jsp*

Update the file details in d1 table and set recount.

- *Regenerate_server2.jsp*

Update file details in d2 table by setting the recount of the table.

- *Regenerate_server3.jsp*

Update the file details in d3 table and set recount to regenerate the damage blocks of file and that would be sent to auditor.

III. System Testing

System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. The testing is paramount and is the final phase. The software, which has been developed, has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of system design, yet it is one of the most important from the efficiency point of view. In the real sense it the phase, which helps to bring out the other phases. Testing is done to make sure that the product does exactly what is supposed to do. Testing is the final verification and validation activity within the organization itself. In the testing stage, we try to achieve the following goals; to affirm the quality of the product, to find and eliminate any residual errors from previous stages, to validate the software as a solution to the original problem, to demonstrate the presence of all specified functionality in the product, to estimate the operational reliability of the system. During testing the major activities are concentrated on the examination and modification of the source code.

A. Software Testing Strategies

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct. The preparation of testing should start as soon as the design of the system starts. To carry out the testing in an efficient manner certain amount of strategic planning has to be done. Any testing strategy must incorporate testing planning, test case design, test execution and the resultant collection and evaluation. The goal will be successfully achieved. There are four steps: Unit Testing, Integration Testing, Validation Testing and Output Testing.

B. Testing of the proposed system

Testing used to find the errors or bugs which found in the software while designing a product. Testing is done for finding faults and weakness in the product which is developed completely. Once the design process of the product has been completed and its ready to be deployed, it must undergo testing to check if its satisfying all the user requirements. It checks the correctness of the product.

Table 1 shows the simple test case samples.

Sl No.	Scenario	Action	Expected Result	Actual Result	Status
1	Run the Application	Homepage should be displayed	Displaying homepage	As Expected	OK
2	User Registration	Registration details has to be displayed	Displaying Registration Page	As Expected	OK
3	User Login	Login details should be displayed	Displaying Login page	As Expected	OK
4	Owner uploads file	Option for uploading file has to be displayed	Displaying file upload option	As Expected	OK
5	Auditor Process	It should view and audit the file	System is displaying the audited file.	As Expected	OK
6	Cloud Process	It should display the files managed by CSP	Displaying file information along with delete option	As Expected	OK
7	Proxy Process	It should login and check for file details	System is displaying file information with failure issues. It sends the same to the auditor.	As Expected	OK
8	File Issues	Auditor should check for file issues	System is displaying issues and notifying the same to the user through email	As Expected	OK
9	Regenerating Process	Proxy should login and get the recovered file.	System is displaying the regenerated files successfully.	As Expected	OK

Fig.1 Simple Test case samples

© IJARET All Rights Reserved

IV. Conclusion

We propose privacy preserving public auditing mechanism for shared data in the cloud. We introduce a TPA to audit the integrity of shared data. We also implement a proxy to regenerate the authenticators. In this paper, we focused upon the implementation and testing aspects of the proposed system. We found out that, system is satisfying most of the requirements and is working as expected.

References

[1] *Boyang Wang, Baochun Li and Hui Li, "Public auditing for shared data with efficient user Revocation in the cloud", IEEE Xplore Digital Library, vol 8, Issue 1, Sep 2015.*

[2] *Kai He, Chuanhe Huang, Kan Yang and Jiaoli Shi, "Identity-preserving public auditing for shared cloud data," in the 23rd IEEE International Symposium on Quality of Service (IWQOS), 2015.*

[3] *P.Divya and B. Sivananthan, "A Privacy-preserving access control with robust data authenticity for cloud group," Journal of Scientific and Computational Intelligence, vol. 2, issue 1, Sep 2015.*

[4] *G. Shreedevi and K.G. Arunkumar, "Survey of public auditing of shared data with multiple third party auditor with efficient user revocation in cloud" Journal of Computer Technology and Applications, vol.6 (2), Mar-Apr 2015.*

[5] *Prof. Sawan Baghel and Prof. Gaurav Saboo, "Efficient Cryptographic algorithms for cloud storage security," Journal of Emerging Technologies in Engineering Research, vol.3, issue 2, Nov 2015.*

[6] *Aparajitha Sain, Parna Dutta, Namrata Dwivedi, Pradnya Chikhale and Vrunda Bhusari, "Enhancing data storage security in cloud computing using PDDS technique," Journal of Advanced Research in Computer Engineering and Technology, vol. 4, issue 2, Feb 2015.*

[7] *Rushikesh P. Dhanokar and Prof. Gitanjali S. Mate, "Auditing of cloud data with privacy preserving using TPA", IOSR Journal of Computer Engineering, 2015.*

[8] *Mr. Santosh P. Jadhav and Prof. B. R. Nandwalkar, "Efficient cloud computing with secure data storage using AES", Journal of Advanced Research in Computer and Communication Engineering, vol. 4, issue 6, June 2105.*

[9] *G. Ranjith, J. Vijaya Chandra, P. Sagarika and B. Prathusha, "Intelligence based Authentication- Authorization and Auditing for secured data storage", Journal of Advanced in Engineering and Technology, vol. 8, issue 4, Aug 2015.*

[10] *Priya Rupeja and Prof. Kalyani Waghmare. "Privacy preserving public auditing and recovery using backup and restore method for secure cloud storage" , Journal of Engineering and Computer Science, volume. 4, issue 1, Jan 2015.*

[11] *Ms. Suvidha R. Sardar and Dr. A. D. Gawande, "Implementation of privacy- preservation in public cloud storage: a Review" , Journal of Advanced Research in Computer Science and Software Engineering, volume 5, issue 4, April 2015.*

[12] *Shruti Batham, Umesh Lilhore and Sini Shibu "Improved HLA based encryption process using fixed size aggregate key generation" , Journal of Modern Trends in Engineering and Research, vol. 2, issue 1, Jan 2015.*

[13] *Mehmet SabirKiraz, Isa Sertkaya and Osmanbey Uzunkol, "An Efficient ID-based message recoverable privacy preserving auditing scheme", in the 13th Annual IEEE Conference on privacy security and trust, 2015.*

[14] *Jianhong Zhang and Xubing Zhao, "Privacy- preserving public auditing scheme for shared data with supporting multi function", Journal of communications, vol. 10, no. 7, July 2015.*

[15] *M. Maha Krishna Jeyanthi, P. Muneeswari, M. Nithya and E. Revathi, "Security and privacy for data sharing in a cloud computing using Ring signature", Journal of Emerging Technology and Innovative Engineering, vol. 1, issue 3, March 2015.*

[16] *Franklin Malugu and K. Suresh Babu, "Public audit of cloud shared data by using efficient privacy preserving scheme", Journal of Scientific Engineering and Research, vol. 3, issue 4, April 2015.*

[17] *Kedar Jayesh Rasal, Dr. S. V. Gumaste and Sandip A. Kahate, "Survey on privacy preserving public auditing techniques for shared data in the cloud", Journal of Engineering Science and Innovative Technology, vol. 4, issue 3, May 2015.*

[18] *B. BanuPriya, V. Sobhana and Prof. Mishmala Sushith, "Concise survey on privacy preserving techniques in cloud", Advanced Research Journal in Science Engineering and Technology, vol. 2, issue 2, Feb 2015.*

[19] *Mr. J. Moses Pushparaj and Ms. K. Rekha, "Enhanced Privacy preserving metadata verification by accomplishing traceability for shared data in cloud", IJAICT, vol. 2, issue 2, June 2015.*

[20] *Pooja Kapadne and Deepak Sharma, "Mechanism for privacy preserving public auditing for shared data in cloud", Journal of Engineering Science and Innovative Technology, vol. 4, issue 5, Sep 2015.*

[21] *Guangyang Yang, Hui Xia, WentingShen, XiuXiu Jiang and Jia Yu, "Public data auditing with constrained auditing number for cloud storage", Journal of security and its applications, vol. 9, issue 9, 2015.*

[22] *Elakkiya B, Savitha S, Vani Parvathi G, Saranya A and Sindhu S, "Public auditing and data dynamics for cloud storage", Journal of Computer Science and Engineering Communications, vol. 3, issue 3, 2015.*

[23] *Remidicherla Rupa, "Auditing outsourced data on cloud using HLA with random masking technique", Journal of Engineering Development and Research, vol. 3, issue 3, 2015.*

[24] *Dhanya Shenoy and N. P.Chawande, "Privacy preserving secure auditing scheme with split cloud storage", Journal of Engineering Trends and Technology, vol. 23, no. 4, May 2015.*

[25] *Kai He, Chuanhe Huang, Haozhou, Jiaolishi, Xiaomao Wang and Feng Dan, "Public auditing for encryption data with client-side deduplication in cloud storage", Wuhan University Journal of Natural sciences, vol. 20, issue 4, August 2015.*

© IJARET All Rights Reserved