ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

Framework for Key Visibility in Cloud Storage Auditing

Sapna R [1], RakeshN [2,] Chandan[3]

Assistant Professor, Dept. of CSE, Sir MVIT, Bangalore, India

B. E Scholar, Dept. of CSE, Sir MVIT, Bangalore, India

B. E Scholar, Dept. of CSE, Sir MVIT, Bangalore, India

ABSTRACT: Providing the security and checking the integrity of the data stored in the public cloud is very important nowadays. Since our project mainly deals with public cloud so, securing the important documents and files are important and we must ensure its integrity. The normal routine followed to provide security to files involves strategies like providing each user of cloud with password as authentication criteria. But this strategy leads to vulnerability of user's password. So, in our project we are introducing and implementing concept of time-based key generation. This process involves generating keys based on the system time and this process is the heart of our entire project. This process in general has two steps. The first step is generating the time-based key and communicating the generated key to authorized user through his registered mail. This is achieved by running the local host server and through SMTP (Simple Mail Transfer Protocol).

KEYWORDS: Key generation; secret key; framework; authentication

I. Introduction

Enabling cloud storage auditing with key exposure resistance is one of the strategy used to achieve the security and integrity aspects of files which are stored in public cloud. Here we are implementing third party administrator (TPA) who administrates the files and documents and checks the integrity of data. Here the administrator has right to view or to download Abby file directly provided, he must concern the owner of that particular file for time-based key.

In this paper we look at how the time-based key generation is implemented in an eclipse setup which involves coding in java. We divide the work into different modules and integrate them to work on local host. We throw a light on using java server pages, various date libraries and its methods, then comparing this strategy with various strategies in later section.

II. Related work

Cong Wang and KuiRen et al Here the data owners or users can remotely store the data in public cloud to facilitate on demand and high quality services but this approach the computational resources were distributer or shared which causes relief of data storage and maintenance at clients side[1]. But, this approach had disadvantage of elimination of physical over storage dependability and security which are basic requirements for any auditing protocol. To overcome this disadvantage the efficient methods were to be designed by data owner itself on behalf of cloud. So, we are introducing an entity between clients and cloud that is, third party administrator, who takes the responsibility of data outsourcing and other aspects.

Giuseppe Ateniese et al Here a model is introduced for provable data possession (PDP)[2]. When the user or owner stored data in untrusted server to ensure that server actually possess the data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs.Here client maintains the constant metadata to verify the proofs. The challenge/response protocol is used to

DOI: 10.15680/IJIRCCE.2017. 0508013

transmit the constant data through network communication. In particular, the overhead at the server is low, as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computations.

ZhuoHao, Sheng Zhong et al Data integrity checking is most Important aspect in cloud auditing[3]. The previously existing protocols achieve this with help of third parties .but in this protocol we are using any third party for checking data integrity. Thus, no private information will be revealed to third party. Through the formal analysis we show the correctness and security of the protocol. After, through theoretical analysis and experimental results, we conclude that the proposed protocol has good performance.

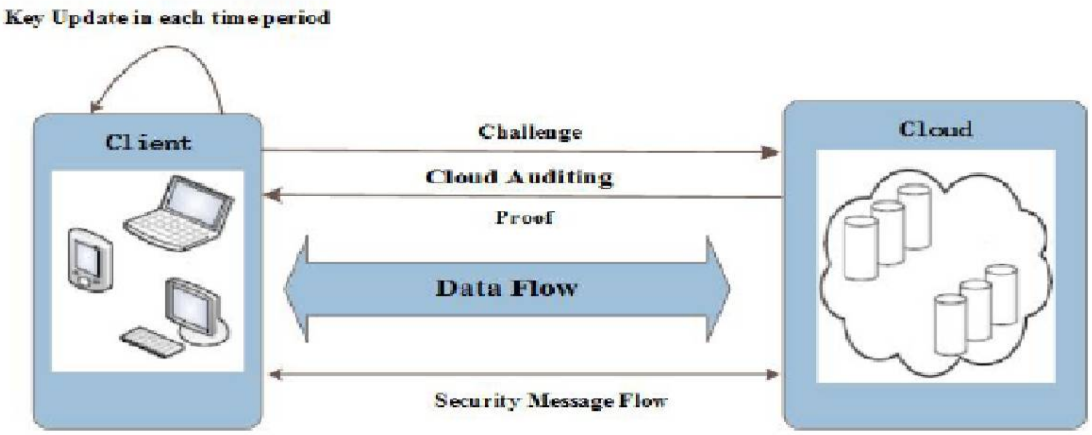
Reza Curtmola et al Storage systems rely on the replication of increase in the durability and availability of data stored in untrusted storage systems[4]. We address this issue through multiple replica provable data possession (MR-PDP). This allows a client that stores t replicas of a file in a storage system to verify through a challenge/response protocol so that, each unique replica can be produced at the time of challenge and the storage system uses t times the storage required to store a single replica.An advantage of MR-PDP is that it can generate further replicas on demand, at little expense, when some of the existing replicas fail.

III. PROPOSED ALGORITHM

Our work has been divided into following modules:

A. Input and output module

B Time key generation and file viewing modules



The input and output modules help the user interact with the software. Time key generation module and mail generation modules are the main modules of this work.

A. Input and Output Module

The input module is in charge of choosing the input data such as user’s credentials and Admin’s data. This modules were designed with HTML and CSS to provide user friendly UI. The user interface should clearly convey the actual process to user.

B. *Time key Generation and file viewing Modules*

This module implements the process of time key generation based on system's time. In file viewing module we implemented how to view each file along with their respective attributes. We used mostly the java server pages (JSP) for allowing users to view in their friendly manner.

One of the important modules is time key generation and file viewing modules. In implementing mail module we are generating mails to registered mail of particular user. This is accomplished by running our local server and transferring through simple mail transfer protocol (SMTP). Java provides various classes like properties etc..to accomplish this mailing mechanism. In the actual implementation of time key module we used basic conditions construct to achieve different keys in different time (time key) whenever particular user requests for an file then he has to generate the time key and provide this as an criteria to view or to download.

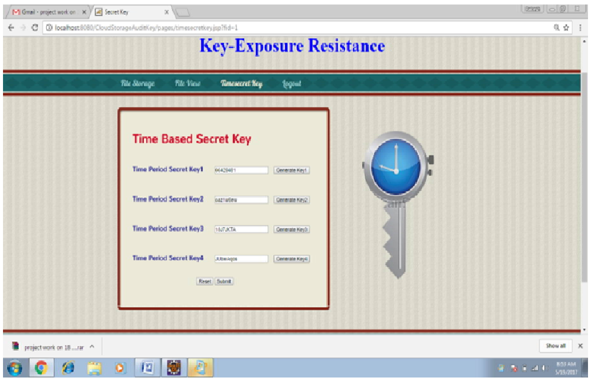
If the authentication fails then our system will report by an error message. This can happen in two ways either user will provide completely wrong key or the valid but expired key.

A. For instance, we are generating four key with an interval of 15 minutes. Like this we can generate the required number of keys with desirable time period.

For mailing the secret key or time key java provides various classes to provide the mechanism of mailing. This is very important to maintain the secrecy of time key.

IV. TIME BASED SECRET KEY GENERATION

"The algorithmic steps we followed for time key generation mechanism is dependent on the system time.
Firstly, generate the set of time keys using the random function (which is the combination of alpha numerical). For instance here we are generating four keys with equal period of life span.



Key-Exposure Resistance

Time Based Secret Key

Time Period Secret Key1: 64429481

Time Period Secret Key2: 6421a8tw

Time Period Secret Key3: 1tJ7JKTA

Time Period Secret Key4: JUbwAqos

Now, the particular key will be active for only its respective assigned time period or interval. If a particular file is requested by user, then the time key which is active at that time will be picked from database and sent to then registered user mail id".

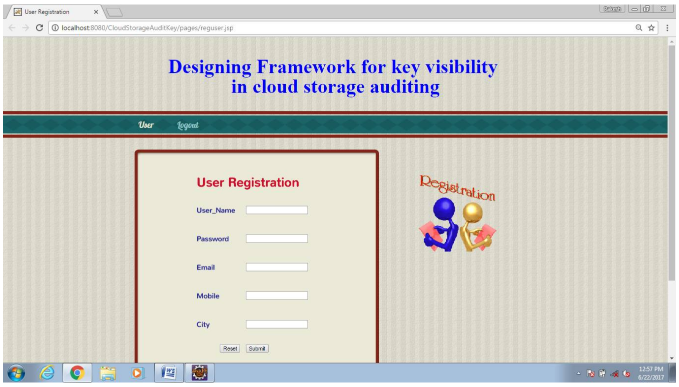
Auditor's view on files:

Here auditor the, Third Party Administrator (TPA) who has the authority and right to check the integrity of data. Provided, he should communicate the owner or user of respective file for the time key.

Auditor has to just provide his permanent secret key and user name as credential. In our paper we came up with the concept where we created only one auditor but in reality we can implement as many auditors depending on cloud size and various characteristics.

V. SIMULATION RESULTS

- USERREGISTRATIONFORM


User Registration

localhost:8080/CloudStorageAuditKey/pages/reguser.jsp

Designing Framework for key visibility
in cloud storage auditing

User Registration

User_Name

Password

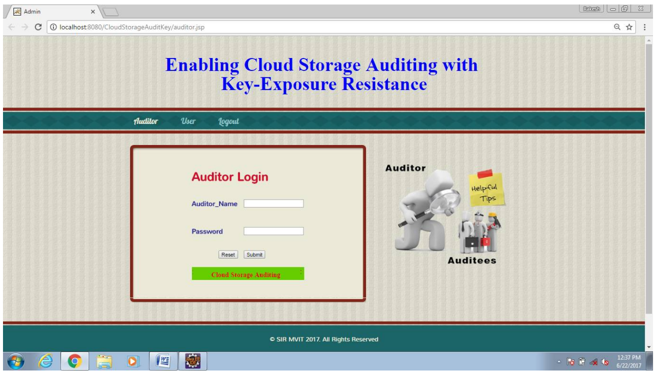
Email

Mobile

City

Here the user needs to fill the form with the intended details. Next the details are validated as below.

- AUTHENTICATING USER


Admin

localhost:8080/CloudStorageAuditKey/auditor.jsp

Enabling Cloud Storage Auditing with
Key-Exposure Resistance

Auditor Login

Auditor_Name

Password

© SIR MVIT 2017. All Rights Reserved

- AUDITOR LOGIN

Admin

localhost:8080/CloudStorageAuditKey/auditor.jsp

Enabling Cloud Storage Auditing with
Key-Exposure Resistance

Auditor Login

Auditor_Name

Password

© SIR MVIT 2017. All Rights Reserved

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

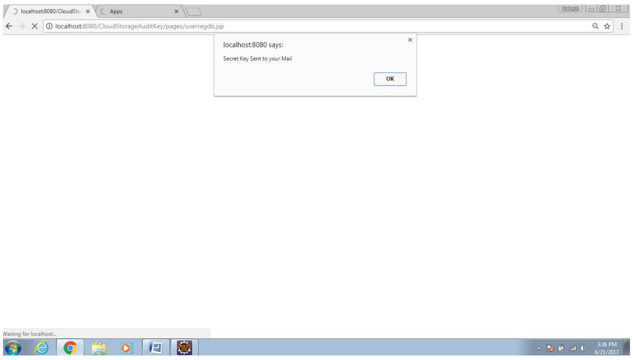
(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 8, August 2017

The auditor’s login page is as above. Next a mailis generated.

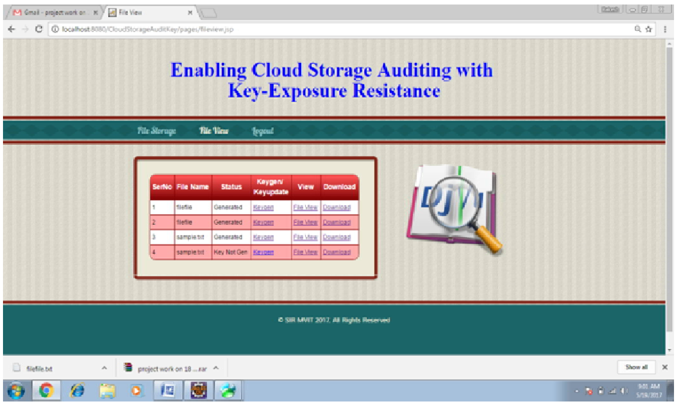
- MAIL GENERATION



localhost:8080 says:

Secret Key Sent to your Mail

- FILES STATUS VIEWING



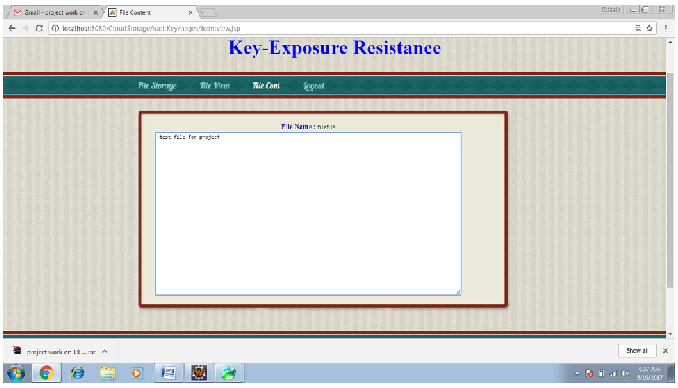
Enabling Cloud Storage Auditing with Key-Exposure Resistance

SerNo	File Name	Status	Keygen/ Keyupdate	View	Download
1	filefile	Generated	Keygen	File View	Download
2	filefile	Generated	Keygen	File View	Download
3	sample.txt	Generated	Keygen	File View	Download
4	sample.txt	Key Not Gen	Keygen	File View	Download

© SIR MVIT 2017. All Rights Reserved

The status of the file can be viewed as above. The file view will be as below.

- FILE VIEW



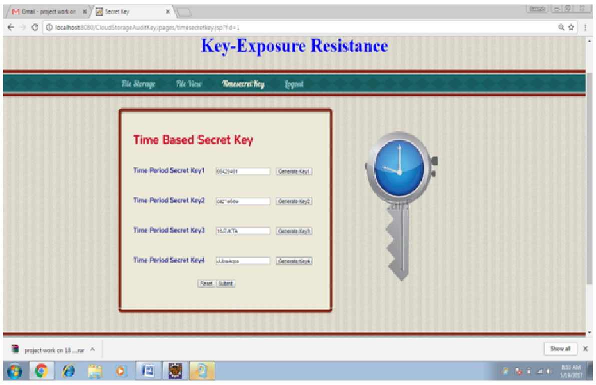
Key-Exposure Resistance

File Name : filefile

test file for project

The key generation is done as below.

- TIME KEY GENERATION



Key-Exposure Resistance

Time Based Secret Key

Time Period Secret Key1 66429481

Time Period Secret Key2 oa21w6ew

Time Period Secret Key3 18J7JKTa

Time Period Secret Key4 uLibwAqoe

VI. CONCLUSION AND FUTURE ENHANCEMENTS

Our aim with this paper has been to provide framework for time based key generation instead of following normal routine of generating only one key for entire life period. So that, even with the exposure of key to public cloud doesn't effects the security algorithm to break down by any intruder or hacker.From our work, we are successful in generating the time-based keys and checked the generation by changing the system time and running the algorithm.

Here we just brought in the concept of designing the framework. By using this we can implement this security protocol in real cloud .By implementing in real cloud this protocol will be synchronized with world clock instead of depending on particular system's time. By implementing in real cloud allows multiple systems as users and runs this protocol to protect their data in public cloud.

REFERENCES

[1] C. Wang , K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.

[3] Y. Zhu, H.G. Ahn, H. Hu, S.S. Yau, H.J. An, and C.J. Hu, "Dynamic Audit Services for Outsourced Storages in Clouds," IEEE Trans. on Services Computing, vol. 6, no. 2, pp. 409-428, 2013.

[4] B. Chen and R. Curtmola, "Auditable Version Control Systems," 2014 Network and Distributed System Security Symposium, 2014.

[5] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Advances in Cryptology-Asiacrypt'08, pp. 90-107, 2008.

[6] K. Yang and X. Jia, "Data Storage Auditing Service in Cloud Computing: Challenges, Methods and opportunities," World Wide Web, vol. 15, no. 4, pp. 409-428, 2012.

[7] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, Vol. 62, No. 2, pp. 362- 375, 2013.

[8] F. Hu, C.H. Wu and J.D. Irwin, "A new forward secure signature scheme using bilinear maps," Cryptology ePrint Archive, Report 2003/188, 2003.

[9] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," Advances in Cryptology-Asiacryp'02, pp. 548-566, 2002.

[10] M. Etemad and A. Kupc¸ ¨ u, "Transparent, distributed, and ¨ replicated dynamic provable data possession," Proc. 11st Applied Cryptography and Network Security. pp. 1-18, 2013.

BIOGRAPHY

Sapna R is an Assistant Professor in the department of Computer Science and Engineering at Sir MVIT, Bangalore. **Rakesh** and **Chandan** are BE Scholars at Sir MVIT, Bangalore.

DOI: 10.15680/IJIRCCE.2017. 0508013