



ENABLING IDENTITY-BASED INTEGRITY AUDITING AND DATA SHARING WITH SENSITIVE INFORMATION HIDING FOR SECURE CLOUD STORAGE

**¹J. Angel Ida Chellam, ²L. Keerthana, ³R. Monica,
¹Assistant Professor, ^{2,3}Student,
^{1,2,3}Department of Information Technology,
^{1,2,3}Sri Ramakrishna Engineering College, Coimbatore.**

ABSTRACT: With cloud storage services, users can remotely store their data to the cloud and realize the data sharing with others. Remote data integrity auditing is proposed to guarantee the integrity of the data stored in the cloud. Encrypting the whole shared file can realize the sensitive information hiding, but will make this shared file unable to be used by others. The proposed scheme is based on identity-based cryptography.

Key words: [Cryptography, Integrity, Auditing, Sanitizer, Proxy, Key Generator.]

1. INTRODUCTION

In some common cloud storage systems such as the electronic health records system, the cloud file might contain some sensitive information. The sensitive information should not be exposed to others when the cloud file is shared. How to realize data sharing with sensitive information hiding in remote data integrity auditing still has not been explored up to now. In order to address this problem, we propose a remote data integrity auditing scheme that realizes data sharing with sensitive information hiding in this paper. In this scheme, a sanitizer is used to sanitize the data blocks corresponding to the sensitive information of the file and transforms these data blocks' signatures into valid ones for the sanitized file. These signatures are used to verify the integrity of the sanitized file in the phase of integrity auditing. As a result, our scheme makes the file stored in the cloud able to be shared and used by others on the condition that the sensitive information is hidden, while the remote

data integrity auditing is still able to be efficiently executed.

2. LITRATURE SURVEY

2.1 K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Comput., vol. 16, no. 1, pp. 69–73, Jan. 2012.

Although cloud computing's benefits are tremendous, security and privacy concerns are the primary obstacles to wide adoption.² Because cloud service providers (CSPs) are separate administrative entities, moving to the commercial public cloud deprives users of direct control over the systems that manage their data and applications.

2.2. G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 598–609.

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server

to verify that the server possesses the original 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.

3. PROPOSED METHODOLOGY

The ID-based RDIC design contains four substances specifically the Trusted key distribution center KGC, cloud server, cloud user, TPA are included in the framework. KGC creates secret keys for every user with their unique identities. The cloud client has a lot of files to be kept on the server without keeping duplicate files, and the cloud server has significant storage space, calculation assets then gives data storage service to the cloud clients. The TPAs work is to play out the information respectability keeping an eye for benefit the cloud client.

The RDIC shows with open absolute status allow anybody to review the integrity of the outsourced information, here let's accept that the third party verifier who has the ability and capability to do the conformation of work.

Modules

- User Key Generation
- Data Upload and Block Split
- Data Auditing and Integrity Checking
- File Download and Recovery

User Key Generation

First the user has to register themselves during the registration process at the site. For this procedure the user will give their very own data and the data will store in the server's database. Now the KGC will generate the secret key to users according to their identities (e.g. name, mail-id, contact number etc.). User gets the provenance to access this application once the key received from KGC.

Data Upload and Block Split

User transfers the files to the server the transferred files are saved into the cloud server. At the point when the user goes to transfer the file in server the information will be encrypted and after that encrypted information will be encoded and now at last encoded record will be splitted into various blocks utilizing dynamic block generation and signatures are stored in file system.

Data Auditing and Integrity Checking

Auditor performs Remote Data Integrity Checking on Cloud Data. Client sends request to auditor to confirm the honesty of the information. Then auditor audits the file by block checking one by one. In Block Checking step: Two signatures are created for Block level Checking.

- From a File System a signature of a block recovers.
- To audit the block a new signature will be generated.

The above two signatures are cross checked for Block level Integrity Checking.

File Download and Recovery

In the event that clients need to download their file, at that point they will send request to cloud, now the cloud will perform block level signature checking process and the block contents are added and requested file will be downloaded in client's framework. Attackers can degenerate data in any of the client's information in cloud. On Data Integrity Checking done by the auditor, The recover process is done automatically when the data gets corrupted. To give the access confidentiality in the cloud blocks will be reallocated and the file system will be updated.

4. SYSTEM SPECIFICATION

HARDWARE REQUIREMENTS:

Hardware	- Pentium
Speed	- 1.1 GHz
RAM	- 1GB
Hard Disk	- 20 GB

Floppy Drive - 1.44 MB
 Key Board - Standard Windows Keyboard
 Mouse - Two or Three Button Mouse
 Monitor - SVGA

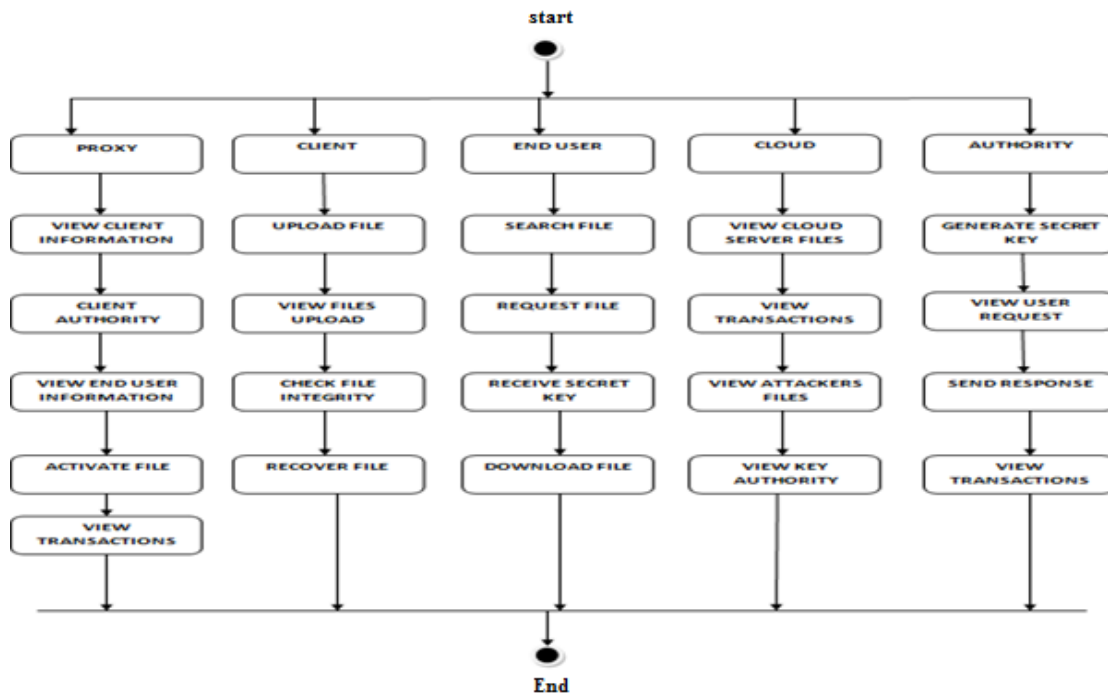
Web Technologie : Html, JavaScript, CSS
 IDE : My Eclipse / NetBeans
 Web Server : Apache Tomcat
 Database : Oracle
 Java Version : J2SDK1.8

SOFTWARE REQUIREMENTS:

Operating System : Windows 7/8/10

Technology : Java and J2EE

ACTIVITY DIAGRAM



5. THE STRATEGIC IMPLEMENTATION PLAN

A strategic implementation plan (SIP) is the document that you use to define your implementation strategy. Typically, it outlines the resources, assumptions, short- and long-term outcomes, roles and responsibilities, and budget. (Later on, we'll show you how to create one.) An SIP is often integrated with an execution plan, but the two are distinct.

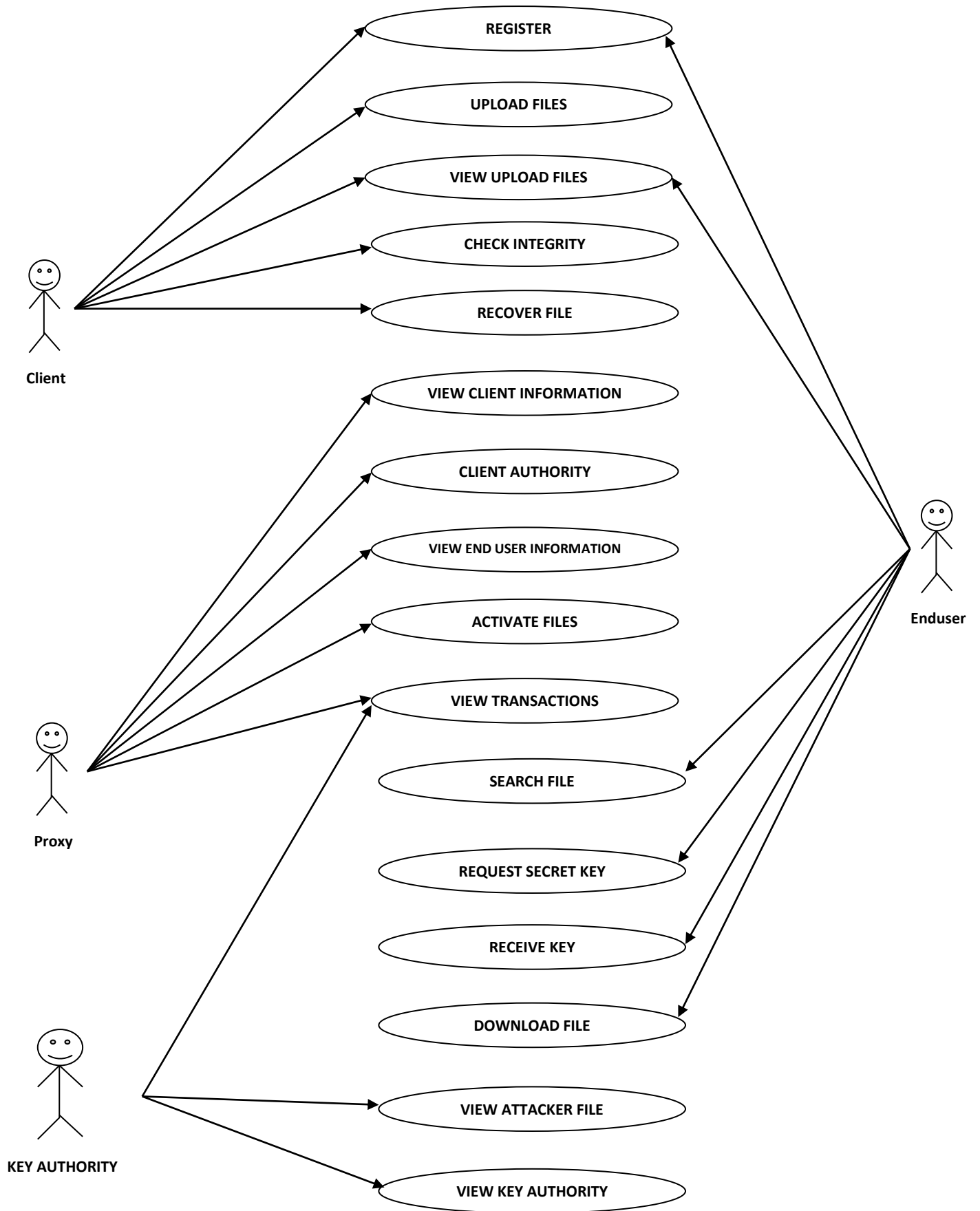
The SIP outlines the activities and decisions necessary to turn the strategic goals into reality, and the execution plan is a schedule of concrete actions and activities to achieve goals and drive success. You can consider your strategy "implemented" once you determine that you have the requisite resources to meet your strategic needs, but you haven't "executed" until you've actually taken

action and achieved objectives. You can read more about the differences between strategy, implementation, and execution in this article by the Harvard Business Review.

The strategic implementation process is often compared to the following activities:

Organizational Planning: Also called organizational design or organizational structure, this is the high-level process of identifying an organization's general operational goals and then building specific strategies to meet them. While strategic planning centers on evaluating an existing organization's strengths and weaknesses and identifying specific goals for improvement, organizational planning is more concerned with how specific tasks, processes, and decisions will flow within the organization.

USECASE DIAGRAM



6. SOFTWARE IMPLEMENTATION



Figure: 1. Home Page (Client and Endusers can upload and search the file)

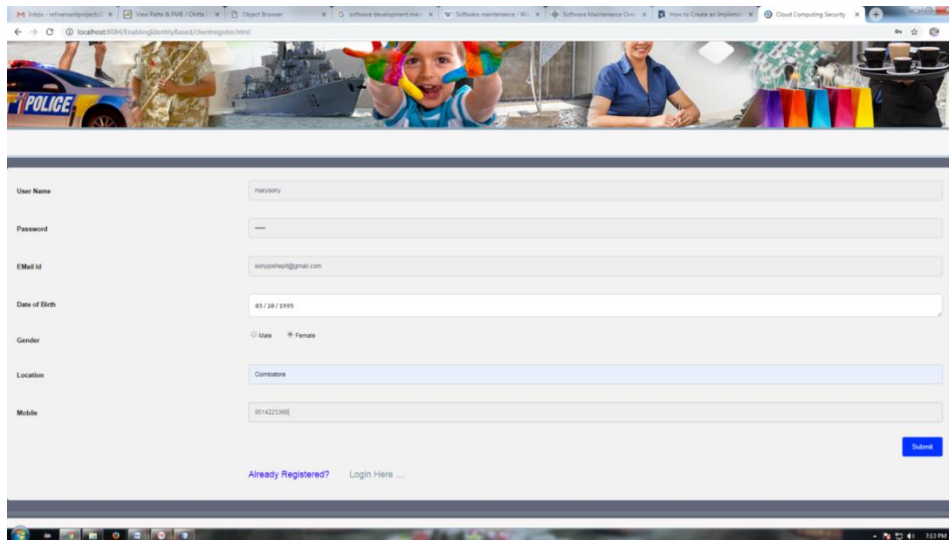


Figure: 2. Client Registration (New clients should register)

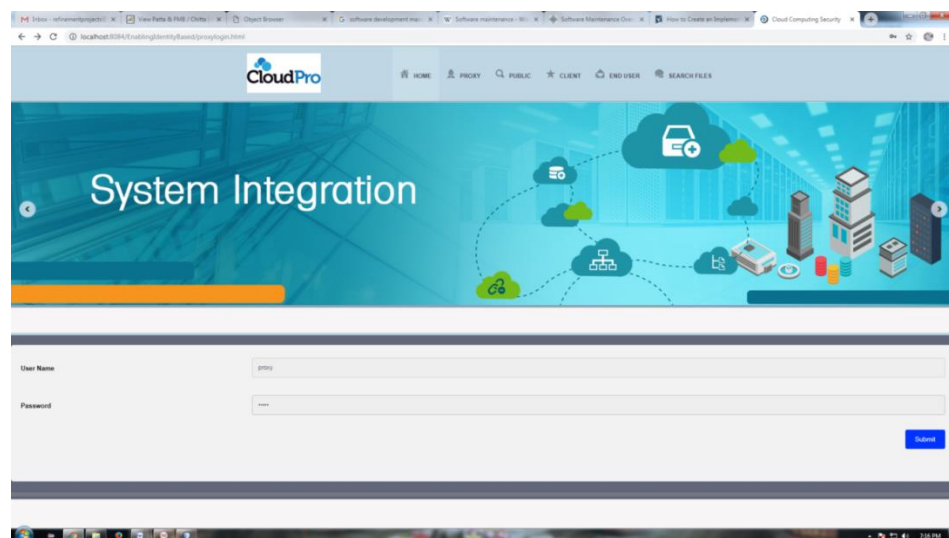


Figure: 3. Proxy Login (proxy have to give approval for the client)

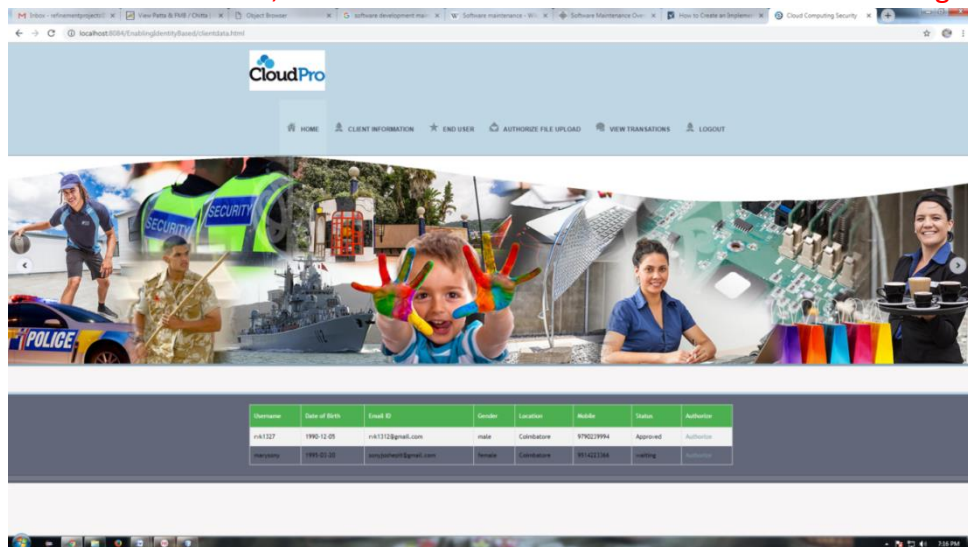


Figure: 4. Client approval (Client must give the approval to the enduser)

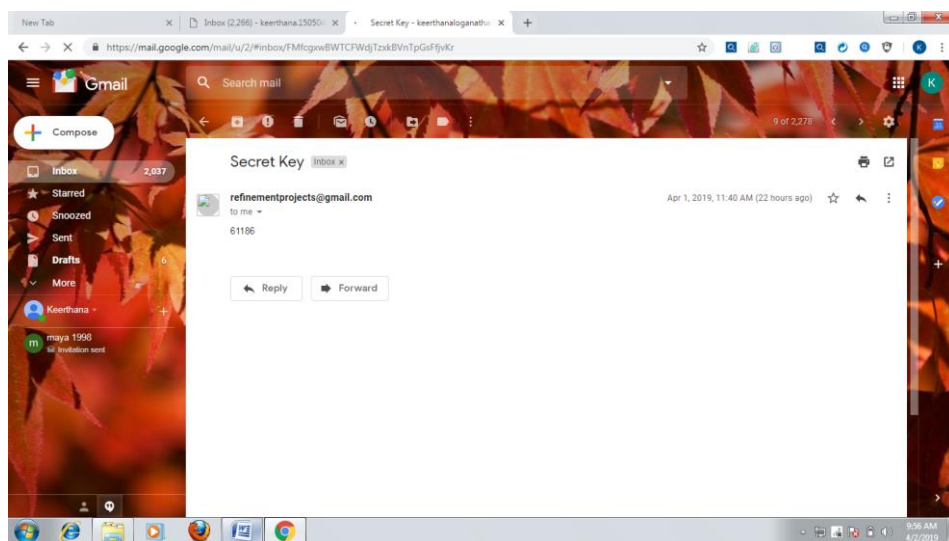


Figure: 5. Secret key (Sent secret key to user mail id)

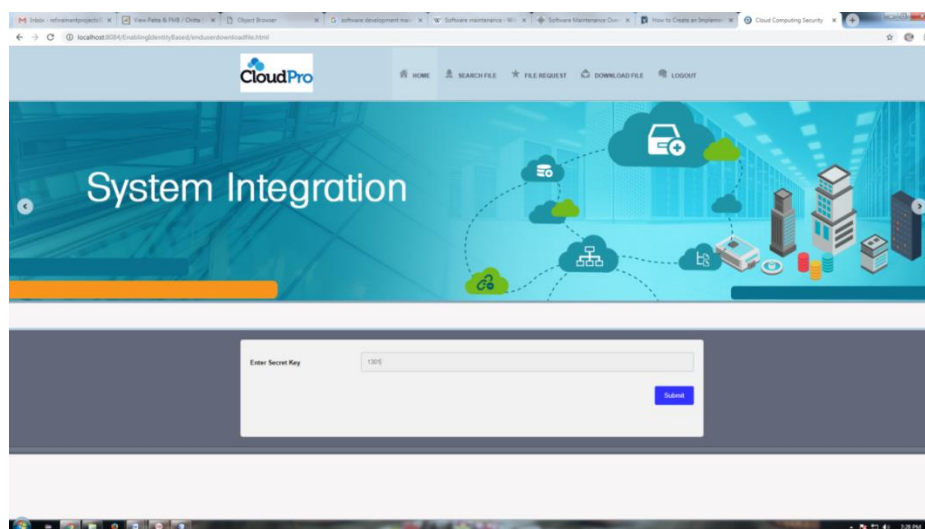
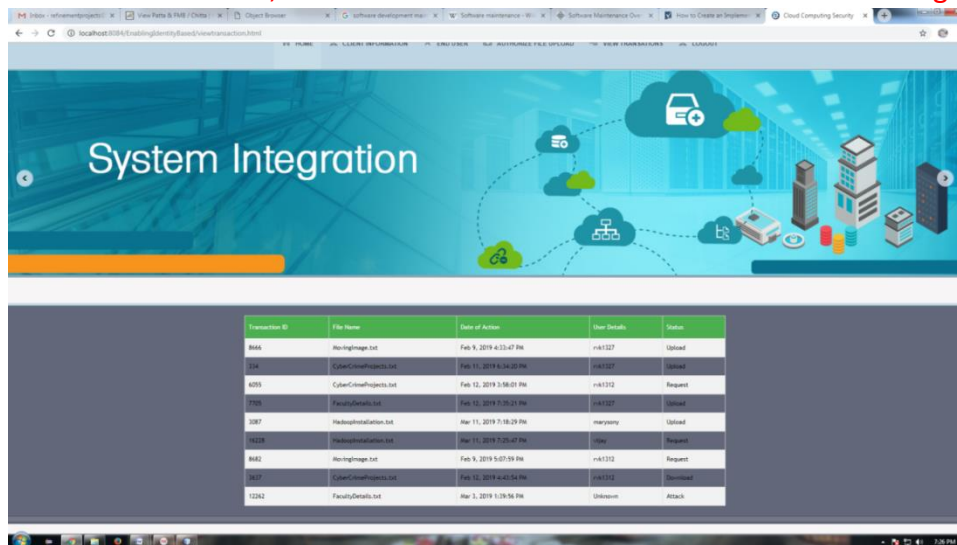
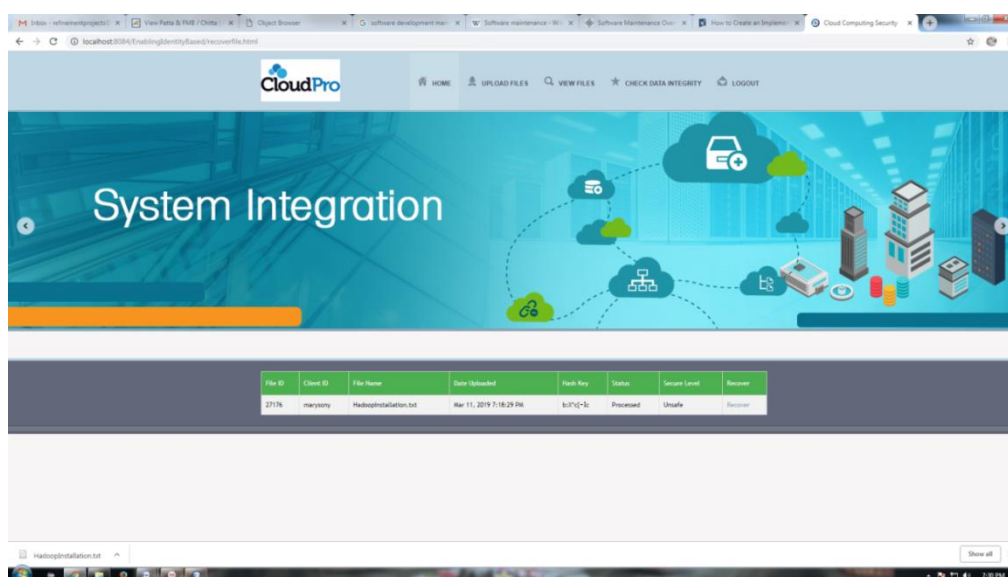


Figure: 6. Download file (Entering secret key and downloading file)



Transaction ID	File Name	Date of Action	User Details	Status
8866	fileimage.txt	Feb 9, 2019 4:25:47 PM	rk1327	Upload
214	CloudCrashProjects.txt	Feb 11, 2019 5:34:30 PM	rk1327	Upload
4289	CloudCrashProjects.txt	Feb 12, 2019 3:58:01 PM	rk1327	Request
7785	HealthDetails.txt	Mar 12, 2019 7:20:29 PM	rk1327	Upload
3287	HealthInstallation.txt	Mar 11, 2019 7:18:29 PM	marjony	Upload
3628	HealthInstallation.txt	Mar 11, 2019 7:25:47 PM	rk1327	Request
8482	fileimage.txt	Feb 9, 2019 4:07:59 PM	rk1327	Request
2607	CloudCrashProjects.txt	Feb 11, 2019 4:41:34 PM	rk1327	Download
12342	HealthDetails.txt	Mar 3, 2019 1:08:56 PM	Unknown	Attack

Figure 7. Data integrity checking (Client can check the integrity of the data)



File ID	Client ID	File Name	Date Uploaded	Hash Key	Status	Server Level	Recovery
271%	marjony	HealthInstallation.txt	Mar 11, 2019 7:18:29 PM	617d1c-3c	Processed	Unsafe	Recover

Figure 8. Data Recovery (client can secure the data)

7. FUTURE ENHANCEMENT

Besides, the remote data integrity auditing is still able to be efficiently executed. The security proof and the experimental analysis demonstrate that the proposed scheme achieves desirable security and efficiency. To evaluate the performance of signature generation and signature verification, we generate the signatures for different number of blocks from 0 to 1000 increased by an interval of 100 in our experiment. The time cost of the signature generation and the signature verification both linearly increases with the number of the data blocks. The time of signature generation ranges from 0.121s to 12.132s.

CONCLUSION

In this paper, we proposed an identity-based data integrity auditing scheme for secure cloud storage, which supports data sharing with sensitive information hiding. In our scheme, the file stored in the cloud can be shared and used by others on the condition that the sensitive information of the file is protected. Besides, the remote data integrity auditing is still able to be efficiently executed. The security proof and the experimental analysis demonstrate that the proposed scheme achieves desirable security and efficiency.

REFERENCE

- [1]. K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Comput.*, vol. 16, no. 1, pp. 69–73, Jan. 2012.
- [2]. G. Ateniese et al., "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 598–609.
- [3]. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur.*, 2007, pp. 584–597.
- [4]. H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, Jul. 2013.
- [5]. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.