PRIVACY IN OFFLOADING DATA FROM MOBILE TO CLOUD

A PROJECT REPORT

Submitted By

SHIVKANTH. B	**31510104096**
SRIRAM. N	**31510104107**
SUGUMAR. K	**31510104112**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SSN COLLEGE OF ENGINEERING

KALAVAKKAM 603110

ANNA UNIVERSITY :: CHENNAI - 600025

March 2014

ANNA UNIVERSITY : CHENNAI 600025

BONAFIDE CERTIFICATE

Certified that this project report titled **"PRIVACY IN OFFLOADING DATA FROM MOBILE TO CLOUD"** is the *bonafide* work of "**Shivkanth. B (31510104096)**, **Sriram. N (31510104107)**, and **Sugumar. K (31510104112)**" who carried out the project work under my supervision.

Dr. Chitra Babu	**Mr. N. Sujaudeen**
Head of the Department	**Supervisor**
Professor,	Assistant Professor,
Department of CSE,	Department of CSE,
SSN College of Engineering,	SSN College of Engineering,
Kalavakkam - 603 110	Kalavakkam - 603 110

Place:
Date:

Submitted for the examination held on. . . . .

Internal Examiner **External Examiner**

ACKNOWLEDGEMENTS

We thank GOD, the almighty for giving us strength and knowledge to do this project.

We would like to thank and extend our deep sense of gratitude to our guide **Mr. N. SUJAUDEEN**, Assistant Professor, Department of Computer Science and Engineering, for his valuable advice and suggestions as well as his continued guidance, patience and support that helped us to shape and refine our work.

Our sincere thanks to **Dr. CHITRA BABU**, Professor and Head of the Department of Computer Science and Engineering, for her words of advice and encouragement and we would like to thank our project Coordinator **Dr. SHEERAZUDDIN**, Associate Professor, Department of Computer Science and Engineering for his valuable suggestions throughout the project.

We express our deep respect to the founder **Dr. SHIV NADAR**, Chairman, SSN Institutions. We also express our appreciation to our Principal **Dr. S. SALIVAHANAN**, for all the help he has rendered during this course of study.

We would like to extend our sincere thanks to all the teaching and non-teaching staffs of our department who have contributed directly and indirectly during the course of our project work. Finally, we would like to thank our parents and friends for their patience, cooperation and moral support throughout our life.

SHIVKANTH. B **SRIRAM. N** **SUGUMAR. K**

ABSTRACT

Many users operate portable devices such as smartphones or tablets that are significantly more limited than desktop computers in terms of on-board memory, processors,useful operating life, and available network bandwidth. Due to mobility and limited wireless network coverage areas, mobile users will likely suffer from transient connectivity such that their constant availability in the system cannot be guaranteed. A mobile user may act as a data owner and decide what access privileges are appropriate for the data that it uploads to the cloud and retains control over; a specific subset of the user population may be identified as having sufficient permission based on unique identities, or users may be assigned various distinguishing attributes that inherently grant permission regardless of the specific identity that assumes them. Outsourcing data to the cloud is beneficial for reasons of economy, scalability, and accessibility, but significant technical challenges remain. Additionally, cloud-based data is increasingly being accessed by resource-constrained mobile devices for which the processing and communication cost must be minimized. To overcome the privacy issues,we propose a technique using attribute and re-encryption based key management with Manager as a trusted entity independent of CSP(Cloud Service Provider).

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

Mobile Cloud Computing binds the cloud computing into mobile environment and it significantly overcomes various obstacles related to performance in terms of battery life , bandwidth and security(reliability and privacy). Cloud based data offers numerous advantages as it enables users of mobile systems to use resources in an on-demand fashion. The rapid explosion of mobile applications and cloud computing support eases the lives of mobile users indirectly. The data that is outsourced to a cloud is appropriate for any type of applications for storage purposes and hence they are disseminated to users. Major problem that is not addressed is that the data stored in the cloud is read by the cloud service provider without the knowledge of the data owner. In addition to that, the cloud provider may not be trusted by all means and there is a considerable chance of data being intercepted by an unauthorised user despite the safety measures undertaken by the cloud provider. It is therefore necessary to use certain encryption algorithms to the data stored in the cloud in order to preserve privacy. It is also advisable that the encryption algorithms that are employed to the data of the data owner must be lightweight in terms of computation and also efficient.

CHAPTER 2

Literature Survey

2.1 A Study on Rijndael Algorithm for providing Confidentiality to Mobile Devices

This is an era of wireless (mobile) communication and computing where mobile devices such as personal digital assistants, smartphones, etc., are being used in place of traditional computers. To secure wireless communication many services and protocols have been developed such as IPSec, SSL/TLS, etc. Heavy computations such as generating RSA keys (large prime numbers) are difficult to generate on mobile devices because the mobile device possesses limited resources. But it can be simplified with the help of smartcards. The paper describes how the smartcards are used to provide content encryption for mobile devices to secure the communication. The well-known Rijndael cryptographic algorithm is used in this concept to secure the communication between the mobile devices or mobile device with any conventional server.

2.2 Improved proxy re-encryption schemes with applications to secure distributed storage

This paper predicts that fast and secure re-encryption will become increasingly popular as a method of managing encrypted file systems. The usefulness of proxy re-encryption as a method to manage access controls on secure files is also demonstrated successfully. The performance measures clearly indicate that this method could be put to good use effectively in practice.

2.3 Trusted data sharing over untrusted cloud storage providers

The off-premises computing paradigm that comes with cloud computing has incurred great concerns on the security of data, especially the integrity and confidentiality of data, as cloud service providers may have complete control on the computing infrastructure that underpins the services. This makes it difficult to share data via cloud providers where data should be confidential to the providers and only authorized users should be allowed to access the data. This work aims to construct a system for trusted data sharing through untrusted cloud providers, to address the above mentioned issue. The constructed system can imperatively impose the access control policies of data owners, preventing the cloud storage providers from unauthorized access and making illegal authorization to access the data.

2.4 A Key-Policy Attribute-Based Broadcast Encryption

According to the broadcast encryption scheme with wide applications in the real world without considering its security and efficiency in the model simultaneously an unbounded, Key-Policy Attribute-Based Broadcast Encryption scheme (KP-ABBE) was proposed by combining with waters dual system encryption, attribute-based encryption and broadcast encryption system. Based on the standard model, the scheme can achieve constant-size public parameters, the public parameters do not impose additional limitations on the functionality of the systems (unbounded) and either a small universe size or a bound on the size of attribute sets avoid to fixed at setup.

2.5 Hybrid Atrribute and Re-encryption based key management for scalable application in clouds.

This paper emphasizes on the fact that a third party user can obtain access to the encrypted data sent by the data owner based on the possession of certain attributes that satisfy an access structure defined in the cloud, rather than the possession of a key that must be disseminated to all interested parties in advance. The required attributes may be determined by a data owner in advance. It is implemented with the help of a Manager who acts as a trusted entity independent of the Cloud Service provider and he jointly cooperates with the data owner in sending the keys .

The technique of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) offers numerous advantages. It allows a user to obtain access to encrypted data in the cloud based on the possession of certain attributes that satisfy an access structure defined in the cloud, rather than the possession of a key that must be disseminated to all interested parties in advance.The requisite attributes may be determined by a data owner in advance; this owner is responsible for generating the user data to be shared, encrypting it and uploading it to the cloud.

2.6 Designing a Hybrid Attribute-Based Encryption Scheme Supporting Dynamic Attributes

This article presents the design of a novel hybrid attribute- based encryption scheme. The scheme is attribute-based, as it allows encrypting under logical combinations of attributes, i.e. properties that users satisfy. It is hybrid, as it combines ciphertext-policy attribute-based encryption (CP-ABE) with location-based encryption (LBE) on the level of symmetric keys. It can efficiently handle dynamic attributes with continuous values, like location, even in resource-constrained settings.

CHAPTER 3

Problem Motivation

Mobile devices namely smartphones and tablets play a significant role in life of a common man and thus they become a necessary tool in providing various services from different applications. But these devices are more limited with respect to desktop computers in terms of on-board memory, processors, and available network bandwidth. Offloading data to the high storage cloud servers invariably improves performance of mobile systems but technical challenges do remain. Moreover, the cloud based data is constantly being accessed by the mobile devices that are resource constrained for which the cost of processing must be minimised.

The security and privacy issues between mobile carriers should be taken care of without consuming much of client resources. So a strong encryption mechanism with least computation time(lightweight) must be employed and effectively implemented

CHAPTER 4

Existing system and Its Limitations

Numerous solutions are provided to exchange encrypted data with a cloud provider in a secure manner, such that the cloud provider is not directly entrusted with key material, but naive schemes often prove difficult to scale. Asymmetric algorithms use two interdependent keys, one to encrypt the data, and the other to decrypt it. The RSA (Rivest, Shamir and Adleman) asymmetric algorithm is widely used in electronic commerce protocols such as SSL, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations. The main drawback of a scheme based on the use of a public key management system such as RSA is that it requires that the data owner provide an encrypted version of data for each recipient that may access it. The technique of Ciphertext-Policy Attribute-Based Encryption (CP-ABE) offers numerous advantages. It allows a user to obtain access to encrypted data in the cloud based on the possession of certain attributes that satisfy an access structure defined in the cloud, rather than the possession of a key that must be disseminated to all interested parties in advance. The requisite attributes may be determined by a data owner in advance; this owner is responsible for generating the user data to be shared, encrypting it and uploading it to the cloud.

4.1 Disadvantages of CP-ABE

For mobile users the communication cost is very high

4.2 Disadvantages of HIBE and CP-ABE

High storage requirements for key material held by users and a greater amount of processing when generating cipher text

CHAPTER 5

Proposed System

The proposed algorithm for key generation, distribution, and usage is now described. It consists of key management techniques that ensure highly secure data outsourcing to the cloud in a highly scalable manner for mobile cloud computing applications. The following improvements are proposed to the basic functions of the original CP-ABE scheme, such that key components have been re-assigned to the various entities in the system model to achieve scalable key management while reducing the mobile user computational and communication workload. A single authority does not generate all key material; the mobile data owner and cloud entity co-operate to jointly compute keys. The cloud provider has insufficient information to decode the user data that it permanently stores. The cloud possesses highly scalable computational ability, unlike a resource-constrained mobile user; a trusted manager also has greater computational resources.

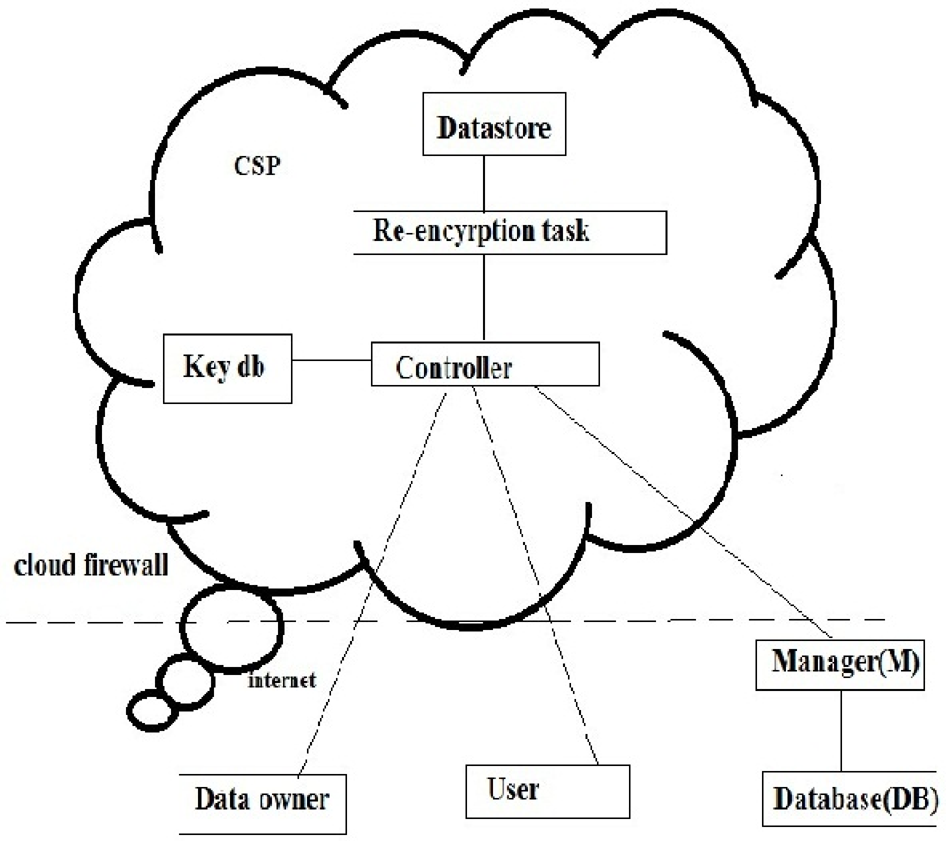


FIGURE 5.1: System Architecture

5.1 Overall Algorithm and Flow

- New user registers.
- Manager allows.
- Manager generates keys
- User chooses the file depending on group and access permissions and encrypts file using key provided by the Manager using AES Rijndael algorithm.

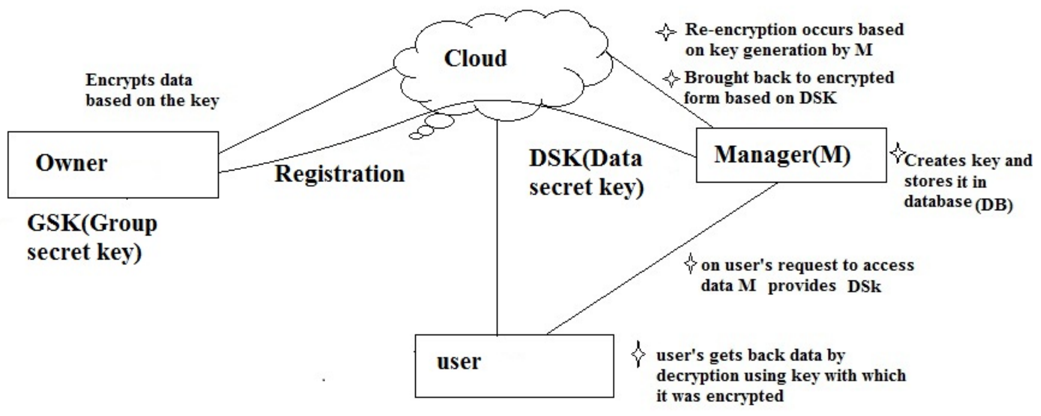

Cloud
Re-encryption occurs based on key generation by M
Brought back to encrypted form based on DSK
Encrypts data based on the key
Owner
Registration
DSK(Data secret key)
Manager(M)
Creates key and stores it in database (DB)
GSK(Group secret key)
on user's request to access data M provides DSk
user
user's gets back data by decryption using key with which it was encrypted

FIGURE 5.2: Flow Diagram

- The file gets uploaded to cloud controller. Cloud controller now requests manager for key(for re-encryption).
- Based on group, access and file name Manager sends a key. Using the key, cloud re-encryption occurs and the re-encrypted data is stored in the cloud.
- Third party user requests the Manager for the file.
- The Manager based on the group and access sends a decryption key.
- The user also requests the owner secret key from the cloud.
- Using the above two keys, the encrypted data is decrypted and plaintext is obtained

CHAPTER 6

Module Split-Up

6.1 User-Data Encryption

The mobile user acting as a data owner of plain-text message m, needs to be encrypted and shared in the cloud with the authorized recipients. The encryption is performed in such a manner that a symmetric cipher such as 256-bit AES(Advanced Encryption Standard) is performed to the entire message so that the plain-text of the data owner is converted to cipher text and stored in the cloud. A manager M acting as a trusted entity within the system is administered under the domain of the users and it is completely independent of cloud service provider(CSP). It is a noble body that is trusted to authorize access to the cloud that contain keys as necessary.

6.2 Cloud Re-Encryption

The Cloud Service provider (cloud controller) is requested to perform a Re-encryption operation on-demand so that its stored cipher text can no longer be decoded using the prior version of the key. The controller requests the manager to provide the re-encryption keys required for performing re-encryption of data. On acceptance by the manager, the key is passed to the Controller and the cipher text is re-encrypted from version 0 to version x. The re-encrypted data is then encoded to ASCII format by Base64 encoding algorithm. ASCII data is easy to be stored and transferred over media (in our case, an Android device). This is to ensure that the data remains intact without modification during transport.

Re-encryption occurs in the cloud based on the key generated by the manager. This dual-encryption scheme is thus a hybrid approach, combining cryptographic techniques, that offers greater flexibility in access control.

1. Additional security mechanism is provided through lightweight encryption algorithm. This is achieved through AES Reijndael algorithm which is more flexible, secure and it also consumes less memory.

2. Re-encryption is done in the cloud to provide additional security to the stored ciphertext in the cloud and it also permits efficient revocation of users.

3. This protocol is designed such that the authorized users can read the data of the data owner based on the possession of the right attributes without any arbitration from the data owner.

4. The responsibility of key generation is divided between data owner and a trusted authority namely Manager and thus the data owner is made free from high computational and other burdens.

5. This proposed protocol is demonstrated on popular mobile and cloud platforms and in addition to that scalability of this protocol is clearly addressed in terms of computational workloads.

6.3 Key Distribution

The manager generates a data secret key(DSK) depending on the attributes and groups of the mobile users. This DSK along with the key generated by the user is used for decrypting the re-encrypted data to encrypted form and based on the key stored in the manager the ciphertext is decrypted back to plaintext for the concerned users.

CHAPTER 7

Encryption

Cloud encryption is a service offered by cloud storage providers whereby data, or text, is transformed using encryption algorithms and is then placed on a storage cloud.

Cloud encryption is the transformation of a cloud service customer’s data into ciphertext. Cloud encryption is almost identical to in host encryption with one important difference the cloud customer must take time to learn about the provider’s policies and procedures for encryption and encryption key management. The cloud encryption capabilities of the service provider need to match the level of sensitivity of the data being hosted.

Because encryption consumes more processor overhead, many cloud providers will only offer basic encryption on a few database fields, such as passwords and account numbers. At this point in time, having the provider encrypt a customer’s entire database can become so expensive that it may make more sense to store the data in-house or encrypt the data before sending it to the cloud. To keep costs low, some cloud providers have been offering alternatives to encryption that dont require as much processing power. These techniques include redacting or obfuscating data that needs to remain confidential or the use of proprietary encryption algorithms created by the vendor.

In the past, many businesses felt comfortable allowing the cloud provider to manage encryption keys, believing that security risks could could be managed through contracts, controls and audits. Over time it has become apparent, however, that cloud providers cannot honor such commitments when responding to government requests for information.

7.1 Rijndael Algorithm

This algorithm is designed to be efficient both in hardware and software across a variety of platforms. Its a block cipher algorithm which works iteratively that has

- Block size: 128 bit (but also 192 or 256 bit)
- Key length: 128, 192, or 256 bit
- Number of rounds: 10, 12 o 14
- Key scheduling: 44, 52 or 60 subkeys having length 32=bits

TYPES:

∗ SubBytes (byte-by-byte substitution using an S-box)

∗ ShiftRows(a permutation, which cyclically shifts the last three rows in the State)

∗ MixColumns(a mixing operation which operates on the columns of the state, combining the four bytes in each column.)

∗ AddRoundKey(bit-by- bit XOR with an expanded key).

7.1.1 Key and Block

Represented with a matrix(array) of bytes with 4 rows and Nk columns, Nk=key length / 32 key of 128 bits= 16 bytes , Nk=4 key of 192 bits= 24 bytes , Nk=6 key of 256 bits= 32 bytes , Nk=8 Block of length 128 bits=16 bytes. It is represented with a matrix array of bytes with

$K_{0,0}$	$K_{0,1}$	$K_{0,2}$	$K_{0,3}$
$K_{1,0}$	$K_{1,1}$	$K_{1,2}$	$K_{1,3}$
$K_{2,0}$	$K_{2,1}$	$K_{2,2}$	$K_{2,3}$
$K_{3,0}$	$K_{3,1}$	$K_{3,2}$	$K_{3,3}$

FIGURE 7.1: Figure representing key.

In_0	In_4	In_8	In_{12}
In_1	In_5	In_9	In_{13}
In_2	In_6	In_{10}	In_{14}
In_3	In_7	In_{11}	In_{15}

FIGURE 7.2: Figure representing block.

4 rows and Nb columns, Nb=block length / 32 and thus for 128 bits Nb=4.Internally, the AES algorithms operations are performed on two-dimensional array of bytes called the State S. S (r,c) denotes the byte in row r and column.

7.1.2 Subbyte Transformation

Byte substitution using a non-linear (but invertible) S-Box (independently on each byte). S-box is represented as a 16x16 array, rows and columns indexed by hexadecimal bits.8 bytes replaced as follows: 8 bytes define a hexadecimal number rc,then sr, c = binary(S-box(r, c)).

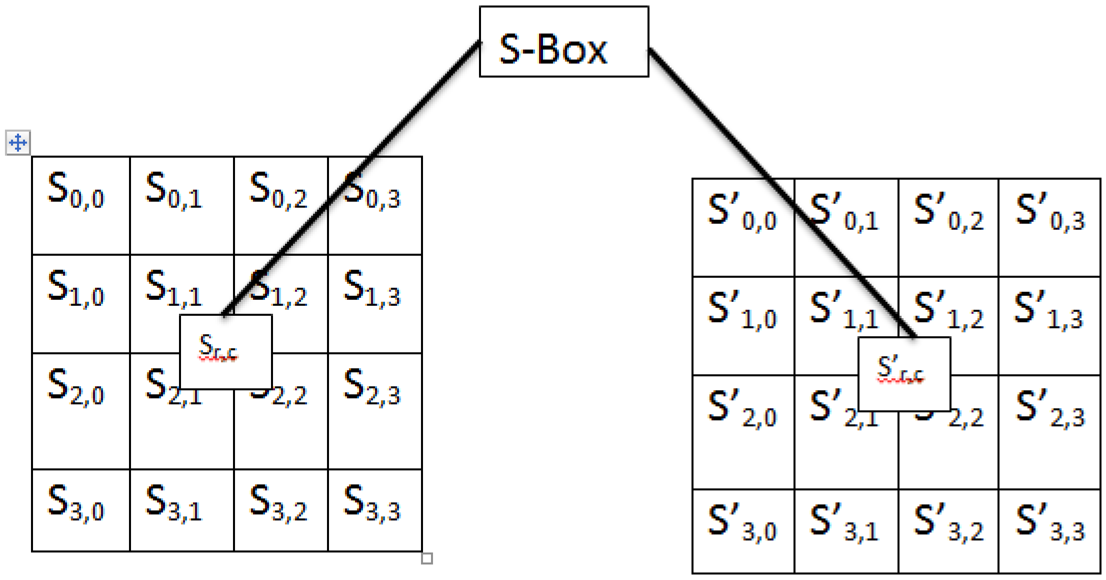


FIGURE 7.3: Figure representing Subbytes.

7.1.3 Shiftrows

The ShiftRows step operates on the rows of the state; it cyclically shifts the bytes in each row by a certain offset. For AES, the first row is left unchanged. Each byte of the second row is shifted one to the left. Similarly, the third and fourth rows are shifted by offsets of two and three respectively.

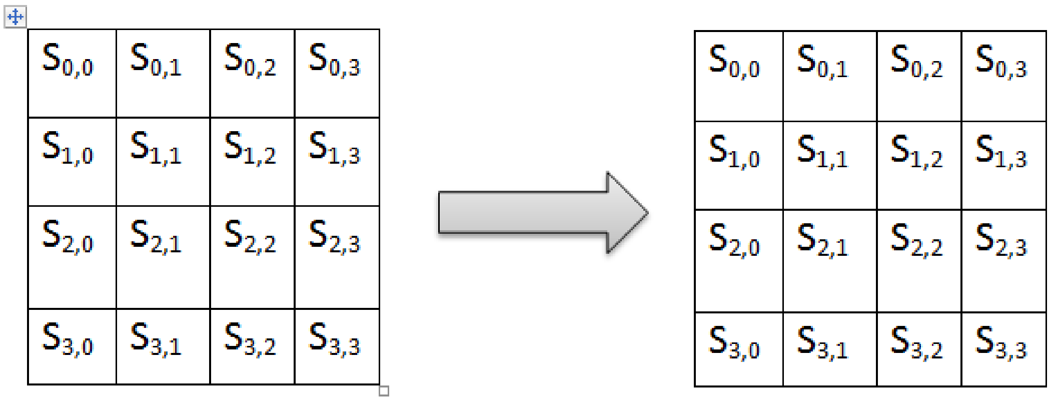

$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$
$S_{0,0}$ $S_{0,1}$ $S_{0,2}$ $S_{0,3}$
$S_{1,0}$ $S_{1,1}$ $S_{1,2}$ $S_{1,3}$
$S_{2,0}$ $S_{2,1}$ $S_{2,2}$ $S_{2,3}$
$S_{3,0}$ $S_{3,1}$ $S_{3,2}$ $S_{3,3}$

FIGURE 7.4: Figure representing Shiftrows.

For blocks of sizes 128 bits and 192 bits, the shifting pattern is the same. Row n is shifted left circular by n-1 bytes. In this way,each column of the output state of the ShiftRows step is composed of bytes from each column of the input state. (Rijndael variants with a larger block size have slightly different offsets). For a 256-bit block, the first row is unchanged and the shifting for the second, third and fourth row is 1 byte, 3 bytes and 4 bytes respectively this change only applies for the Rijndael cipher when used with a 256-bit block, as AES does not use 256-bit blocks. The importance of this step is to avoid the columns being linearly independent, in which case, AES degenerates into four independent block ciphers.

7.1.4 Mixcolumns

Interpret each column as a vector of length 4.Each column of State is replaced by another column obtained by multiplying that column with a matrix in a particular field(Galois Field). Matrix multiplication is composed of multiplication and addition of the entries, and here the multiplication operation can be defined as this: multiplication by 1 means no change, multiplication by 2 means shifting to the left, and multiplication by 3 means shifting to the left and then performing XOR with the initial unshifted value.

- After shifting, a conditional XOR with 0x1B should be performed if the shifted value is larger than 0xFF. (These are special cases of the usual multiplication in GF(28).)
- Addition is simply XOR.

7.1.5 AddRoundKey

In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation(). The subkey is combined with the state. For each round, a subkey is derived from the main key and each subkey is the same size as the state. The subkey is added by combining each byte of the state with the corresponding byte of the subkey using bitwise XOR.

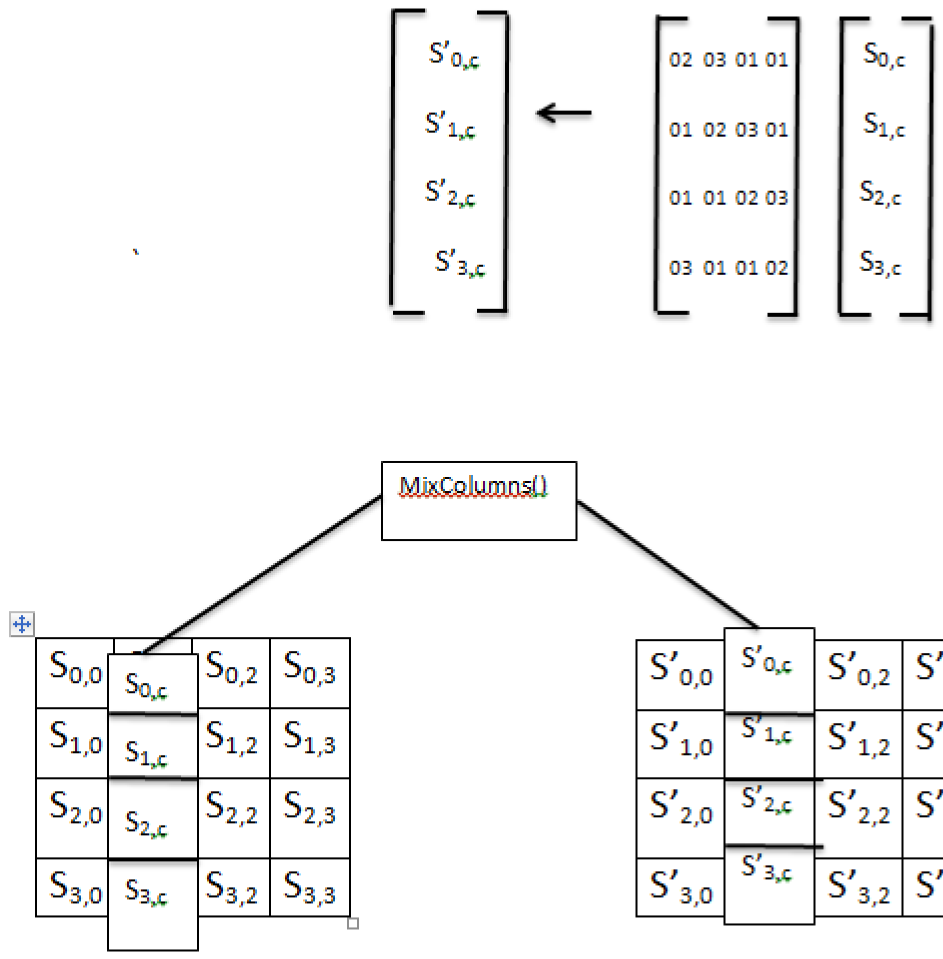


FIGURE 7.5: Figure representing Mixcolumns.

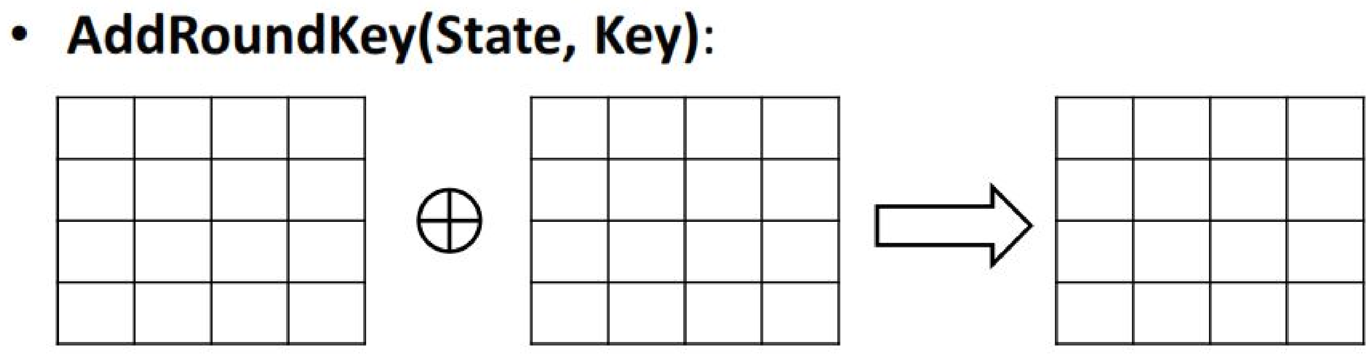


FIGURE 7.6: Figure representing AddRoundKey.

7.2 Base64 Encoding

Base64 is a group of similar binary-to-text encoding schemes that represent binary data in an ASCII string format by translating it into a radix-64 representation. The term Base64 originates from a specific MIME content transfer encoding. Base64 encoding schemes are commonly used when there is a need to encode binary data that needs to be stored and transferred over media that is designed to deal with textual data. This is to ensure that the data remains intact without modification during transport. Base64 is commonly used in a number of applications including email via MIME, and storing complex data in XML.

Base 64 encodes the input data three bytes at a time. Each block of three input bytes is encoded to create a block of four printable characters. The three bytes are ordered into a 24 bit value, starting from the most significant bit of Byte 0, and ending with the least significant bit of Byte 2. The bits are then arranged as a set of four 6 bit numbers, N0 to N3. The first 6 bits form N0, the next 6 form N1 etc. Each 6 bit number has a range 0 to 63.

The second stage is to convert numbers N0N3 into ASCII characters, C0C3. This is done according to the following table:

In addition, the MIME specification states that encoded data should have a CRLF character pair inserted after every 76 characters (or less) of encoded data.The original data can be any length, not necessarily a multiple of 3. This means that the last block of binary data could be 1, 2 or 3 bytes long.

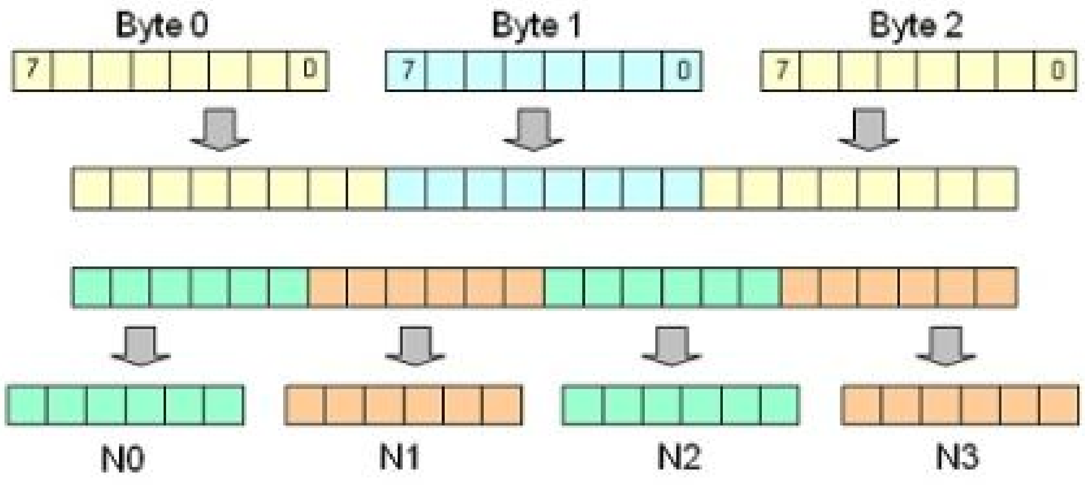
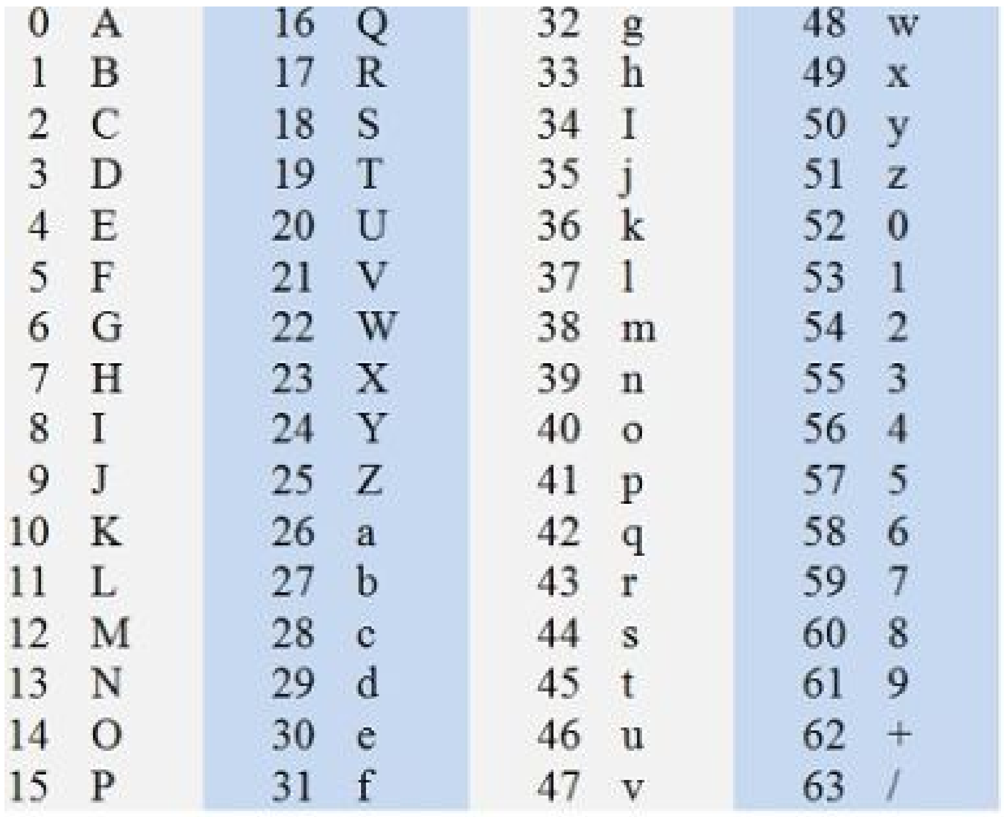


FIGURE 7.7: Figure representing Byte Process.

To code the final block, we add zeros to the final block to make it a multiple of 3, and convert it to 4 characters as usual. However,we indicate the length of the block in the following way:If the final block has a length of 1 byte, the encoded characters consist of C0, C1, followed by 2 = characters (C2 and C3 contain no useful information anyway). If the final block has a length of 2 bytes, the encoded characters consist of C0, C1, C2, followed by a single = character. If the final block has a length of 3 bytes, the encoded characters consist of C0, C1, C2, C3 in then normal way.

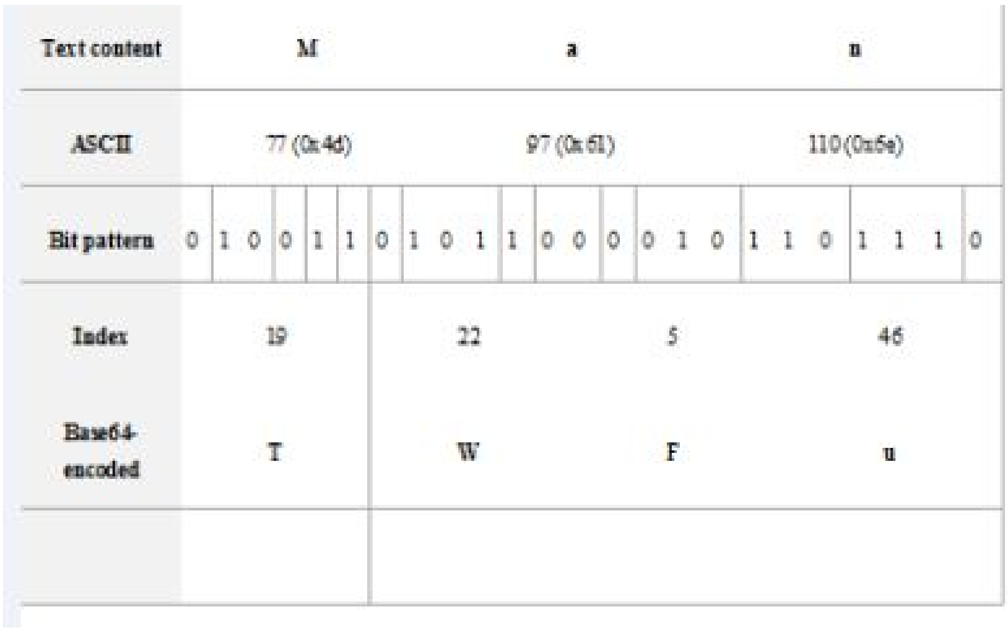
The encoded value of Man is TWFu. Encoded in ASCII, the characters M, a, and n are stored as the bytes 77, 97, and 110, which are the 8-bit binary values 01001101,01100001 and 01101110. These three values are joined together into a 24-bit string, producing 010011010110000101101110. Groups of 6 bits (6 bits have a maximum of 26=64 different binary values) are converted into individual numbers from left to right (in this case, there are four numbers in a 24-bit string), which are then converted into their corresponding Base64 character values. As



0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	I	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

FIGURE 7.8: Figure representing Character Mapping.

this example illustrates, Base64 encoding converts three octets into four encoded characters.



Text content	**M**								**a**								**n**							
ASCII	77 (0x4d)								97 (0x61)								110 (0x6e)							
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Index	19						22						5						46					
Base64-encoded	**T**						**W**						**F**						**u**					

FIGURE 7.9: Figure representing example of Base64.

CHAPTER 8

Software Requirement Specification

8.1 Overall Description

8.1.1 Product Perspective

Many users will operate portable devices such as smart phones or tablets that are significantly more limited than desktop computers in terms of on-board memory processors, useful operating life, and available network bandwidth. Due to mobility and limited wireless network coverage areas, mobile users will likely suffer from transient connectivity such that their constant availability in the system cannot be guaranteed. A mobile user may act as a data owner and decide what access privileges are appropriate for the data that it uploads to the cloud and retains control over; a specific subset of the user population may be identified as having sufficient permission based on unique identities, or users may be assigned various distinguishing attributes that inherently grant permission regardless of the specific identity that assumes them. A highly scalable system is envisioned where users may, number potentially in the thousands or millions. Continuous arbitration by a single data owner during all transactions is impractical. So, the manager is a trusted authority within the system and is administered under the domain of the users in question; it is completely independent of the CSP. A controller administrates access through external client interfaces. Requests, including data uploads and downloads, are made over the reliable but insecure medium of the Internet; a wireless packet data infrastructure serves to bridge mobile users.

8.1.2 Product Features

Data with a cloud provider in a secure manner, such that the cloud provider is not directly entrusted with key material, but naive schemes often prove difficult to scale. For instance, the main drawback of a scheme based on the use of a public key management system such as RSA is that it requires that the data owner provide an encrypted version of data for each recipient that may access it. If user data is encrypted with a single key, then that key must be shared with all authorized users, which carries a high traffic cost especially if this obligation rests on a mobile data owner. Users may join and leave the authorized user set frequently, leading to constant key regeneration and re-distribution through additional communication sessions to handle user revocation; in a highly scalable system, such events may occur at relatively high frequency. Wireless communication, however, is expensive and results in rapid battery drain, especially when transmitting. Data should ideally be stored in the cloud in encrypted form so that the cloud provider cannot access it. This notion is dependent on the keys being securely managed by an entity outside of the providers domain. The difficulty arises when new users join the system, and existing ones leave, necessitating new keys to be generated. The encrypted data should ideally be transformed such that it may be unlocked with new keys, without an intermediate decryption step that would allow the cloud provider to read the plaintext; this process is known as data re-encryption. Our Proposed System consists of key management techniques that ensure highly secure data outsourcing to the cloud in a highly scalable manner for mobile cloud computing applications.

Mobile User uploads a data to cloud in encrypted form. So that the cloud service provider may be unable to view the file. But from the user end, if the secret key gets compromised then the data get revealed to everyone. So in order to overcome that manager provides re-encryption keys based on attributes to the cloud controller. When controller receives that, cloud controller will re-encrypt that data. When user wants to access the data they have to get the decryption keys from manager. So by using the re-encryption key they will get the first version of cipher text. And by using their own secret keys they will get the plain text.

8.1.3 Operating Systems

8.1.3.1 Software Requirements

1. Windows Operating System 2000 and Above
2. JDK 1.6
3. Tomcat Server 6.0
4. My SQL 5.5
5. Android mobile
6. XAMPP

8.1.3.2 Hardware Requirements

1. Hard Disk: 10GB and Above
2. RAM: 512MB and Above

3. Processor: Pentium III

8.1.4 Cloud Platform

eyeOS is a web desktop following the cloud computing concept that seeks to enable collaboration and communication among users. It is mainly written in PHP, XML, and JavaScript. It is a cloud application platform with a web-based desktop interface. Commonly called a cloud desktop because of its unique user interface, eyeOS delivers a whole desktop from the cloud with file management, personal management information tools, collaborative tools and with the integration of the clients applications.

8.1.5 Design and Implementation Constraints

8.1.5.1 Constraints in Analysis

1. Determination of the Involved Classes
2. Determination of the Involved Objects
3. Determination of the Involved Actions
4. Determination of the Require Clauses
5. Global actions and Constraint Realization

8.1.5.2 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical at one. It is rather straightforward to transform the developed hierarchical model into a bipartite, at model, consisting of classes on the one hand and at relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation purposes. There is no identity or functionality associated with a at relation. A at relation corresponds with the relation concept of entity-relationship modelling and many object oriented methods.

8.1.6 System Features

An improvement is made over a traditional attribute based encryption scheme, such that responsibility over key generation is divided between a mobile data owner and a trusted authority; the owner is relieved of the highest computational and messaging burdens. The provider is unable to read stored data; authorized users may do so based on qualification through possession of the right attributes without arbitration by the data owner. The protocol is designed to be efficient for resource constrained mobile users by delegating computation and requests to a cloud provider or trusted authority, where appropriate, without compromising security. Suppose that Alice is a mobile user that acts as the self elected data owner Uo of plaintext message m, which is user data that is desired to be encrypted and shared in the cloud with other authorized users. Regardless, the length of the message does not impact the size or number of encryption keys required. In the case of message segmentation, the same pre-computed keys may be applied to all segments.

8.2 External Interface Requirements

8.2.1 User Interfaces

All the contents in the project are implemented using Graphical User Interface (GUI) in Java through Javafx. Every conceptual part of the projects is reflected using the Javafx with Java.System gets the input and delivers through the GUI based.

8.3 ISDN

You can connect your AS/400 to an Integrated Services Digital Network (ISDN) for faster, more accurate data transmission. An ISDN is a public or private digital communications network that can support data, fax, image, and other services over the same physical interface. Also, you can use other protocols on ISDN, such as IDLC and X.25.

8.3.1 Software Interfaces

This software is interacted with the TCP/IP protocol, Socket and listening on unused ports. Server Socket and listening on unused ports and JDK 1.6. This software is also interacted with the SMTP protocol, sending and receiving on SMTP protocol.

8.3.2 Communication Interfaces

TCP/IP protocol,SMTP.

8.4 Other Nonfunctional Requirements

8.4.1 Performance Requirements

We conducted several experiments to test the performance of BEES in the existence of errors in distance estimations and angle measurements. To account for errors in distance measurements due to the irregularity of signal We also conducted an experiment to compare the localization accuracy achieved by BEES against other RSS-based localization techniques.

8.4.2 Safety Requirements

1. The software may be safety-critical. If so, there are issues associated with its integrity level

2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.

3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.

4. There is little point in producing perfect code in some language if hardware and system software (in widest sense) are not reliable.

5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.

6. Systems with different requirements for safety levels must be separated.

7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

8.4.2.1 Security Requirements

Do not block the some available ports through the windows firewall.

8.4.2.2 Software Quality Attributes

- Functionality: are the required functions available, including Interoperability and security.
- Reliability: maturity, fault tolerance and recoverability
- Usability: how easy it is to understand, learn, and operate the software system

- Efficiency: performance and resource behavior.
- Maintainability: Maintaining the software.
- Portability: The software can easily be transferred to another environment.

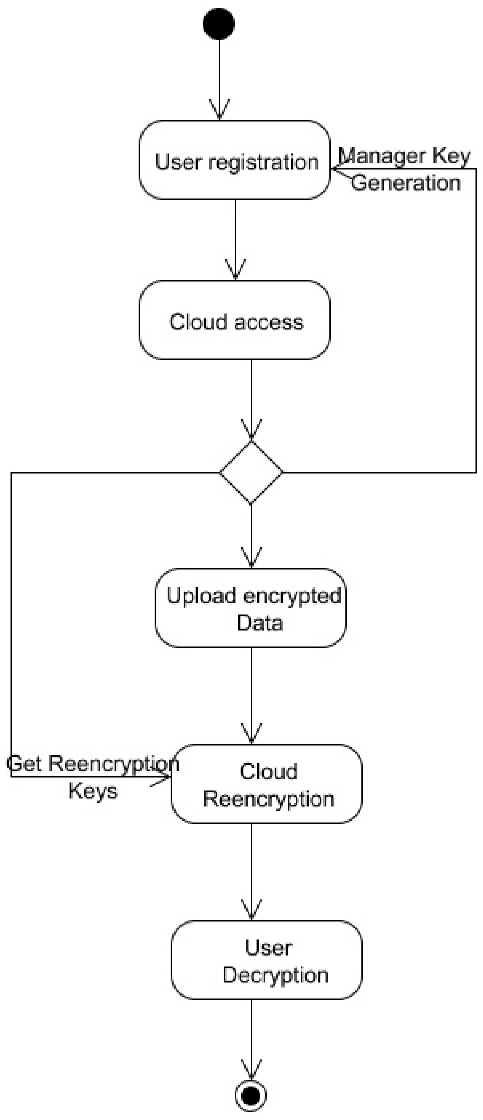


FIGURE 8.1: Activity Diagram

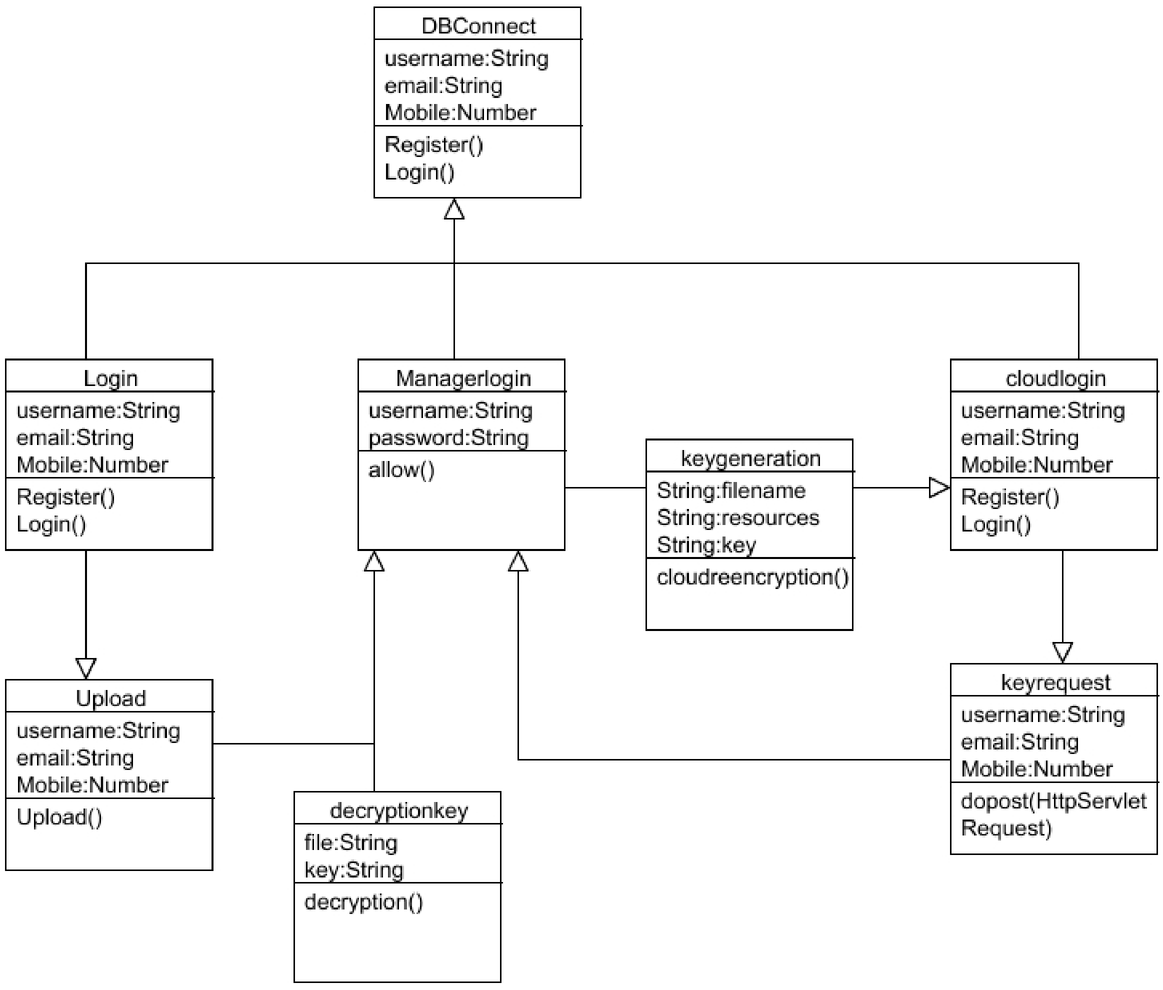


FIGURE 8.2: Class Diagram

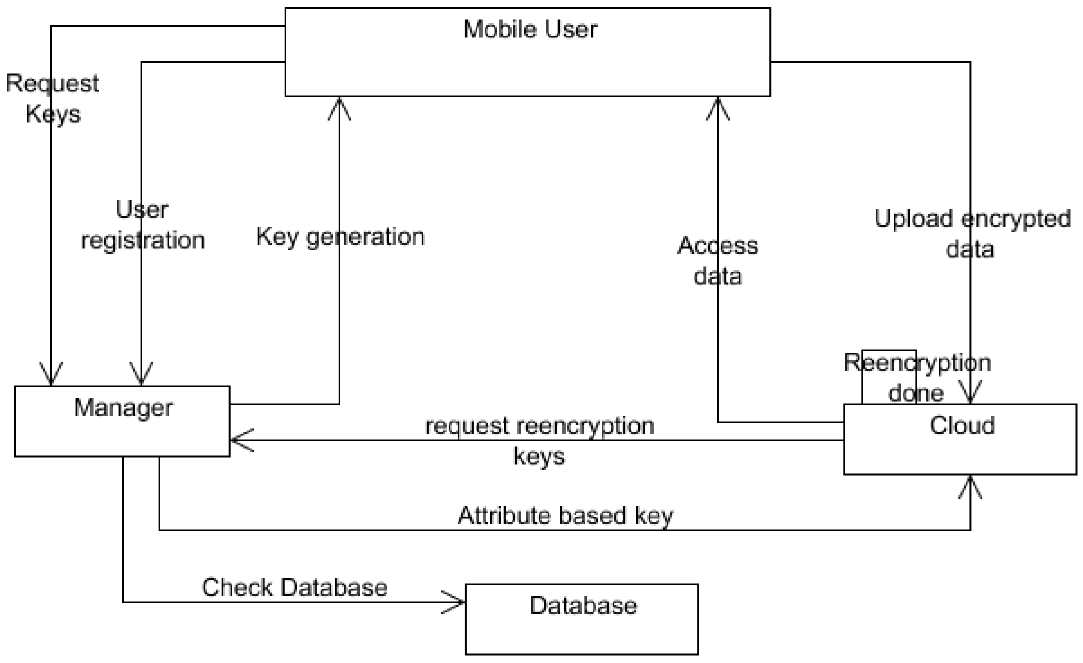


FIGURE 8.3: Collaboration Diagram

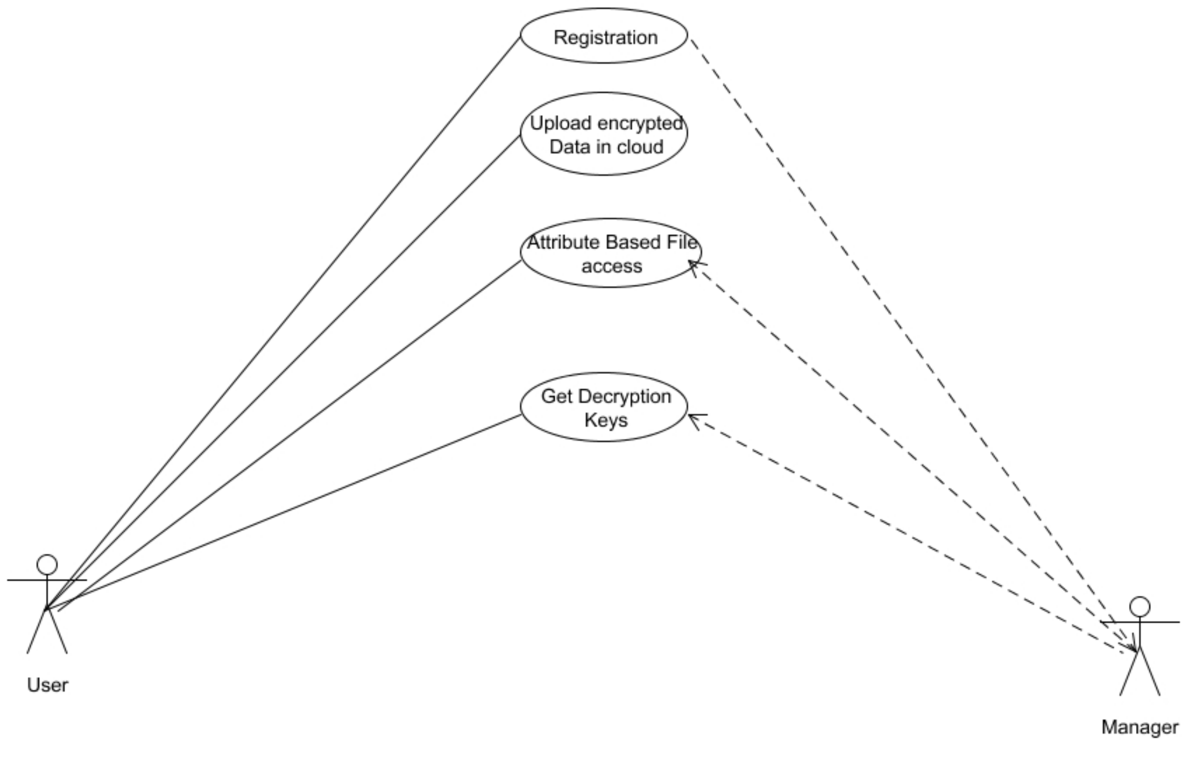
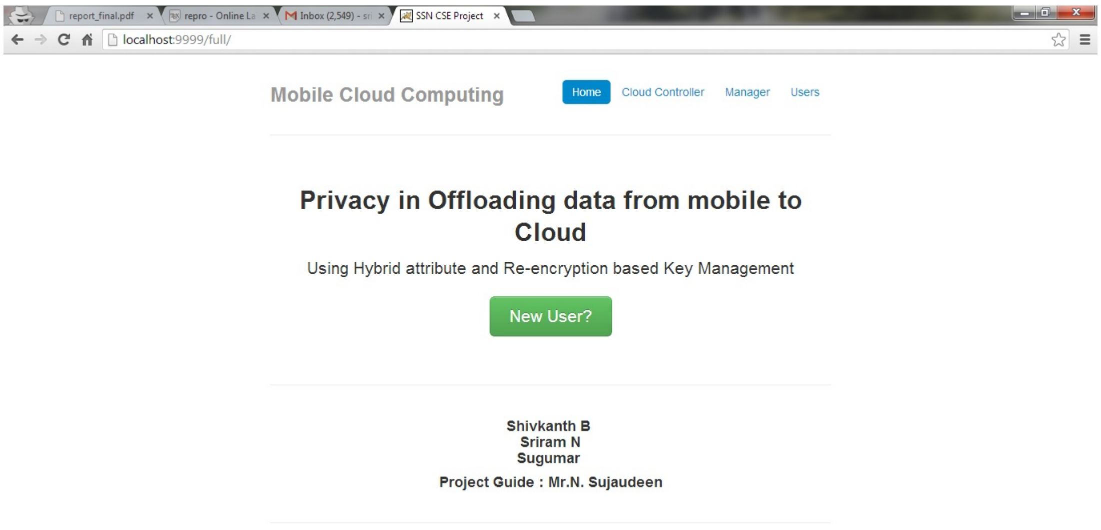


FIGURE 8.4: Usecase Diagram

CHAPTER 9

RESULTS

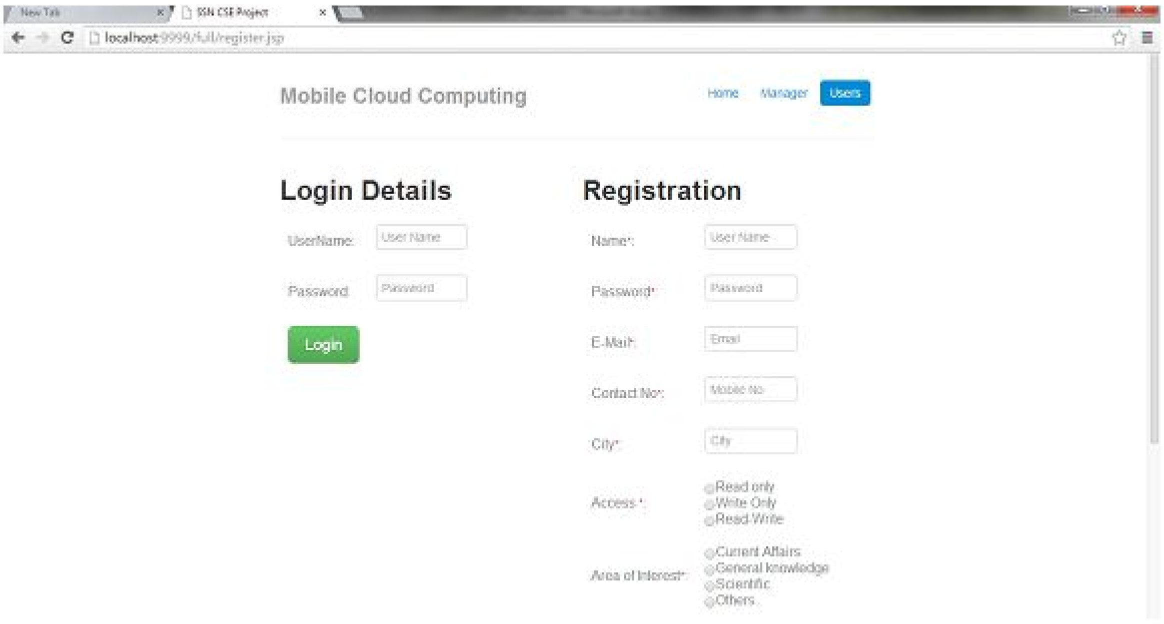


Privacy in Offloading data from mobile to Cloud

Using Hybrid attribute and Re-encryption based Key Management

Shivkanth B
Sriram N
Sugumar
Project Guide : Mr.N. Sujaudeen

FIGURE 9.1: Homepage



localhost:9999/full/register.jsp

Mobile Cloud Computing

Home Manager Users

Login Details

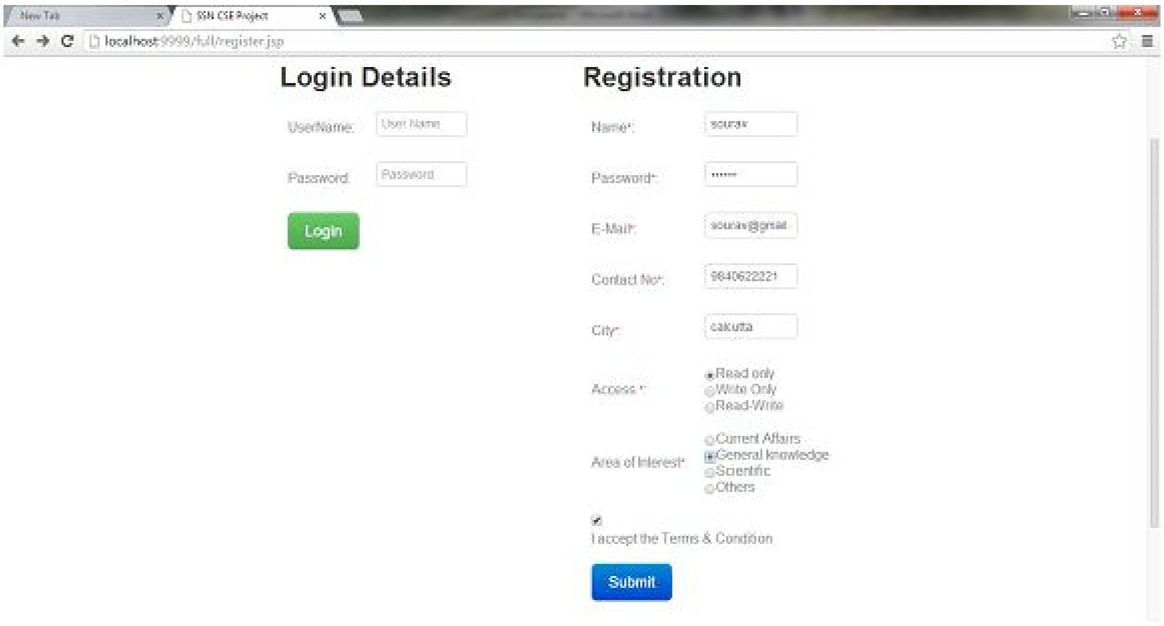
UserName:	User Name
Password:	Password

Login

Registration

Name*:	User Name
Password*:	Password
E-Mail*:	Email
Contact No*:	Mobile No
City*:	City
Access *:	○ Read only ○ Write Only ○ Read-Write
Area of Interest*:	○ Current Affairs ○ General knowledge ○ Scientific ○ Others

FIGURE 9.2: User Registration page



localhost:9999/full/register.jsp

Login Details

UserName:	User Name
Password:	Password

Login

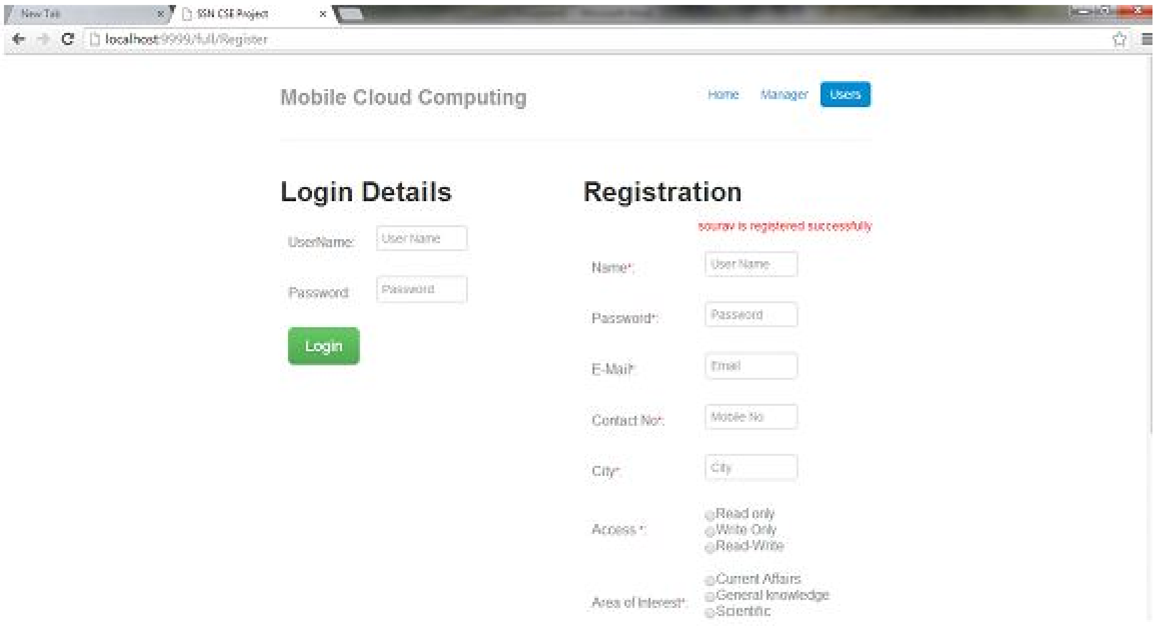
Registration

Name*:	sourav
Password*:	••••••
E-Mail*:	sourav@gmail.
Contact No*:	9840622221
City*:	calcutta
Access *:	[✓ Read only] ○ Write Only ○ Read-Write
Area of Interest*:	○ Current Affairs [✓ General knowledge] ○ Scientific ○ Others

✓ I accept the Terms & Condition

Submit

FIGURE 9.3: New user sign-up


localhost:9999/full/Register

Mobile Cloud Computing

Home Manager Users

Login Details

UserName: User Name

Password: Password

Login

Registration

sourav is registered successfully

Name*: User Name

Password*: Password

E-Mail*: Email

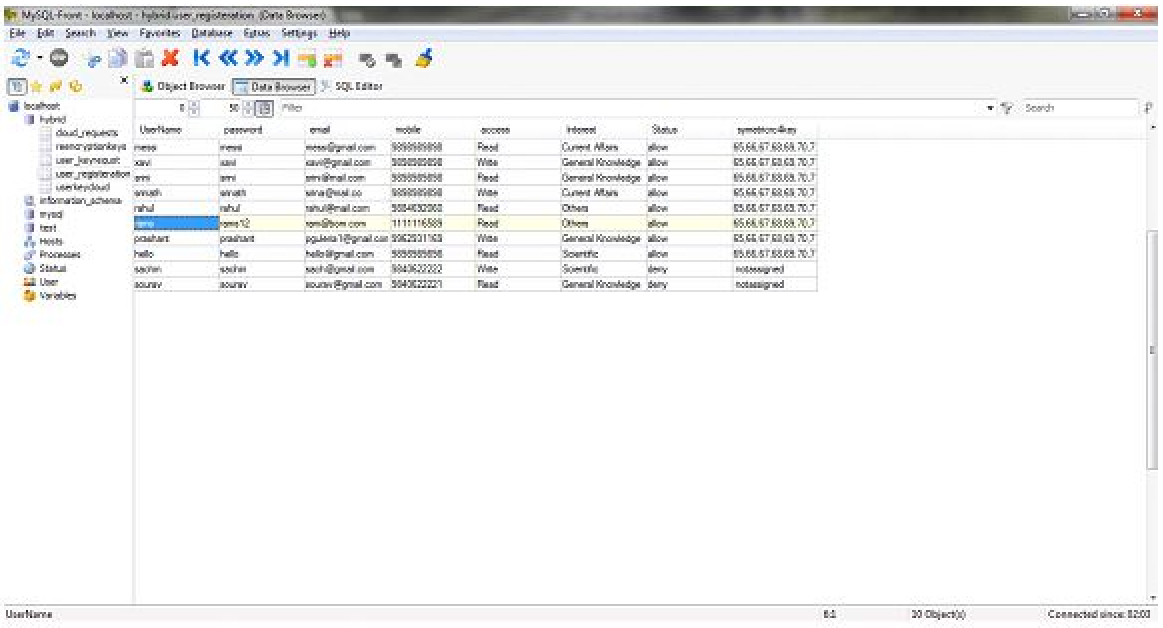
Contact No*: Mobile No

City*: City

Access *: Read only / Write Only / Read-Write

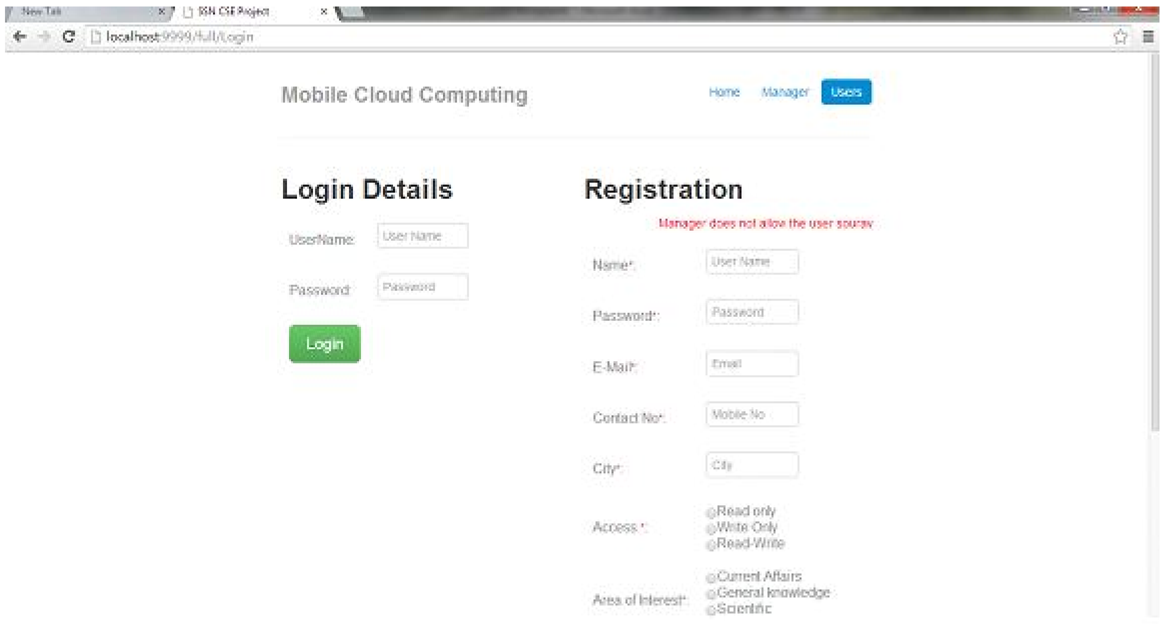
Area of Interest*: Current Affairs / General knowledge / Scientific

FIGURE 9.4: Registration notification


MySQL-Front - localhost - hybrid.user_registeration (Data Browser)

UserName	password	email	mobile	access	interest	Status	symetrickey
messi	messi	messi@gmail.com	9898989898	Read	Current Affairs	allow	65,66,67,68,69,70,7
xavi	xavi	xavi@gmail.com	9898989898	Write	General Knowledge	allow	65,66,67,68,69,70,7
ami	ami	ami@mail.com	9898989898	Read	General Knowledge	allow	65,66,67,68,69,70,7
amrath	amrath	sana@mail.co	9898989898	Write	Current Affairs	allow	65,66,67,68,69,70,7
rahul	rahul	rahul@mail.com	9884632060	Read	Others	allow	65,66,67,68,69,70,7
[illegible]	rama12	ram@bom.com	1111116589	Read	Others	allow	65,66,67,68,69,70,7
prashant	prashant	pgulesta1@gmail.com	9962931163	Write	General Knowledge	allow	65,66,67,68,69,70,7
hello	hello	hello@gmail.com	9898989898	Read	Scientific	allow	65,66,67,68,69,70,7
sachin	sachin	sach@gmail.com	9840622222	Write	Scientific	deny	notassigned
sourav	sourav	sourav@gmail.com	9840622221	Read	General Knowledge	deny	notassigned

FIGURE 9.5: User registration table(database)



New Tab | SSN CSE Project

localhost:9999/full/Login

Mobile Cloud Computing

Home | Manager | Users

Login Details

UserName:	User Name
Password:	Password

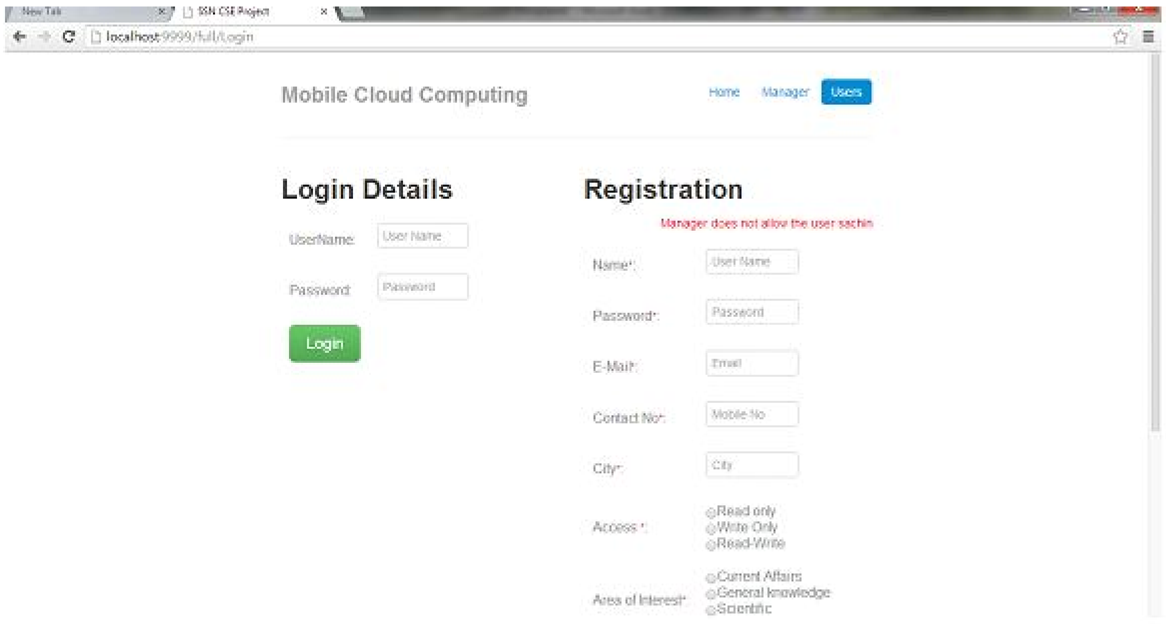
Login

Registration

Manager does not allow the user sourav

Name*:	User Name
Password*:	Password
E-Mail*:	Email
Contact No*:	Mobile No
City*:	City
Access *:	Read only Write Only Read-Write
Area of Interest*:	Current Affairs General knowledge Scientific

FIGURE 9.6: Manager on disapproval of user 1



New Tab | SSN CSE Project

localhost:9999/full/Login

Mobile Cloud Computing

Home | Manager | Users

Login Details

UserName:	User Name
Password:	Password

Login

Registration

Manager does not allow the user sachin

Name*:	User Name
Password*:	Password
E-Mail*:	Email
Contact No*:	Mobile No
City*:	City
Access *:	Read only Write Only Read-Write
Area of Interest*:	Current Affairs General knowledge Scientific

FIGURE 9.7: Manager on disapproval of user 2

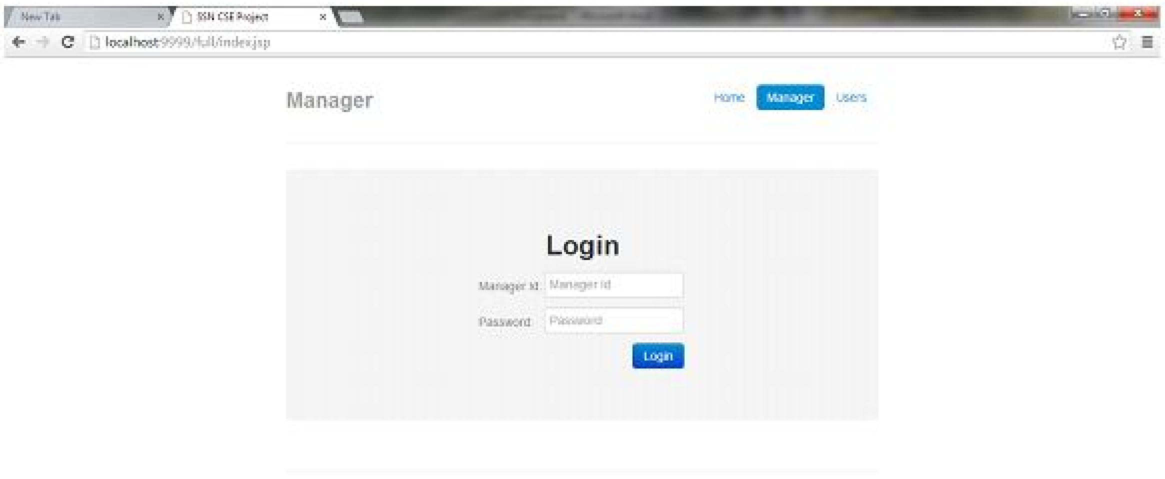


FIGURE 9.8: Manager homepage

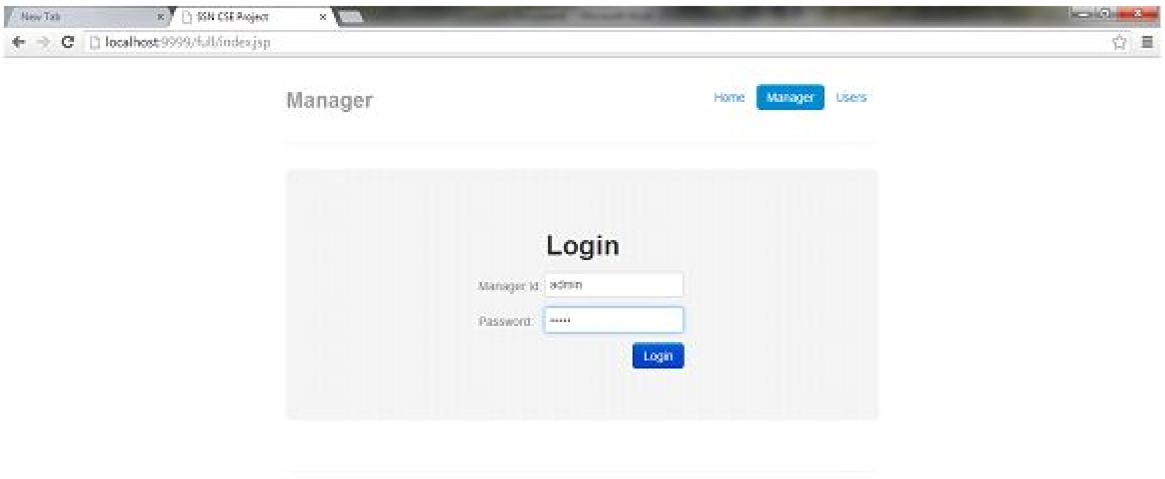
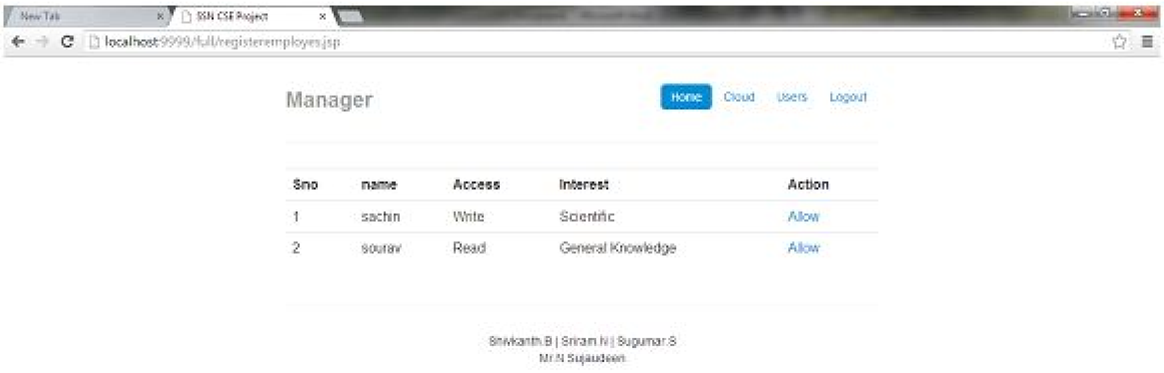


FIGURE 9.9: Manager login



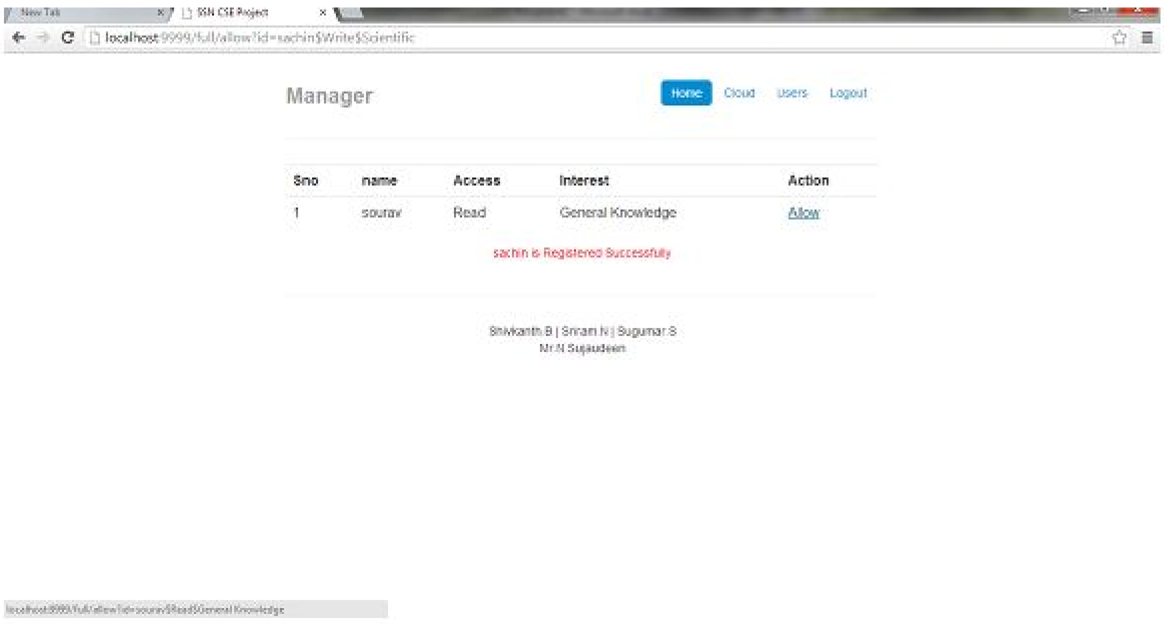
localhost:9999/full/registeremployes.jsp

Manager

Sno	name	Access	Interest	Action
1	sachin	Write	Scientific	Allow
2	sourav	Read	General Knowledge	Allow

Shivkanth.B \| Sriram.N \| Sugumar.S
Mr.N Sujaudeen

FIGURE 9.10: Manager approving users



localhost:9999/full/allow?id=sachin$Write$Scientific

Manager

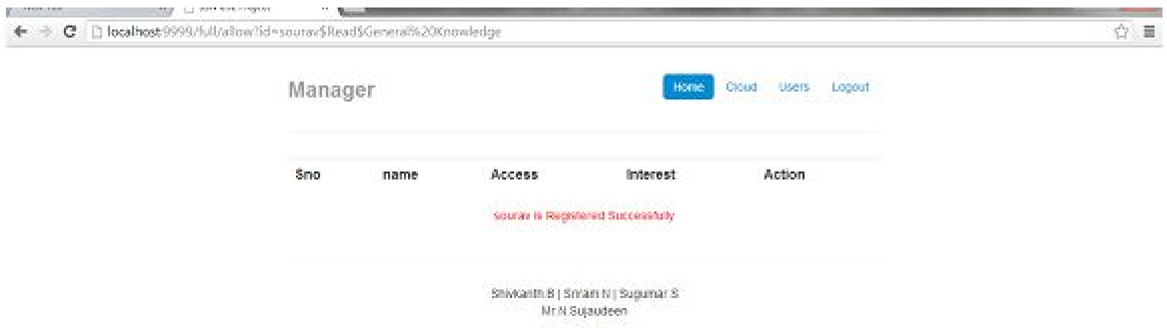
Sno	name	Access	Interest	Action
1	sourav	Read	General Knowledge	Allow

sachin is Registered Successfully

Shivkanth.B \| Sriram.N \| Sugumar.S
Mr.N Sujaudeen

localhost:9999/full/allow?id=sourav$Read$General Knowledge

FIGURE 9.11: Manager on approval of user 1



localhost:9999/full/allow?id=sourav$Read$General%20Knowledge

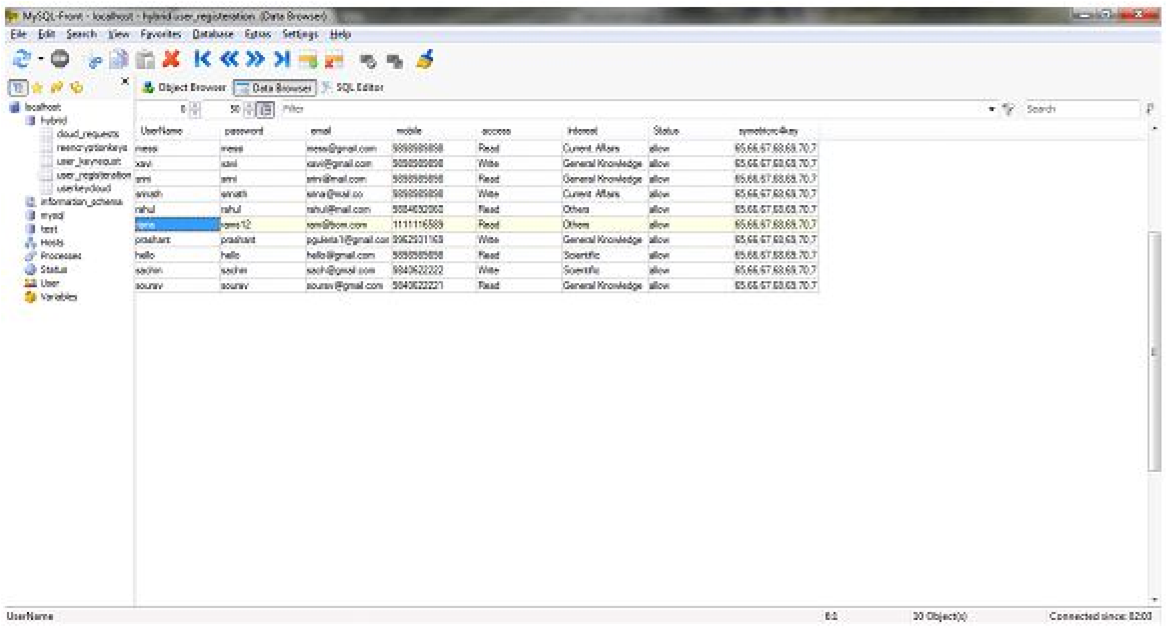
Manager

Sno	name	Access	Interest	Action

sourav is Registered Successfully

Shivkanth B | Sriram N | Sugumar S
Mr.N Sujaudeen

FIGURE 9.12: Manager on approval of user 2

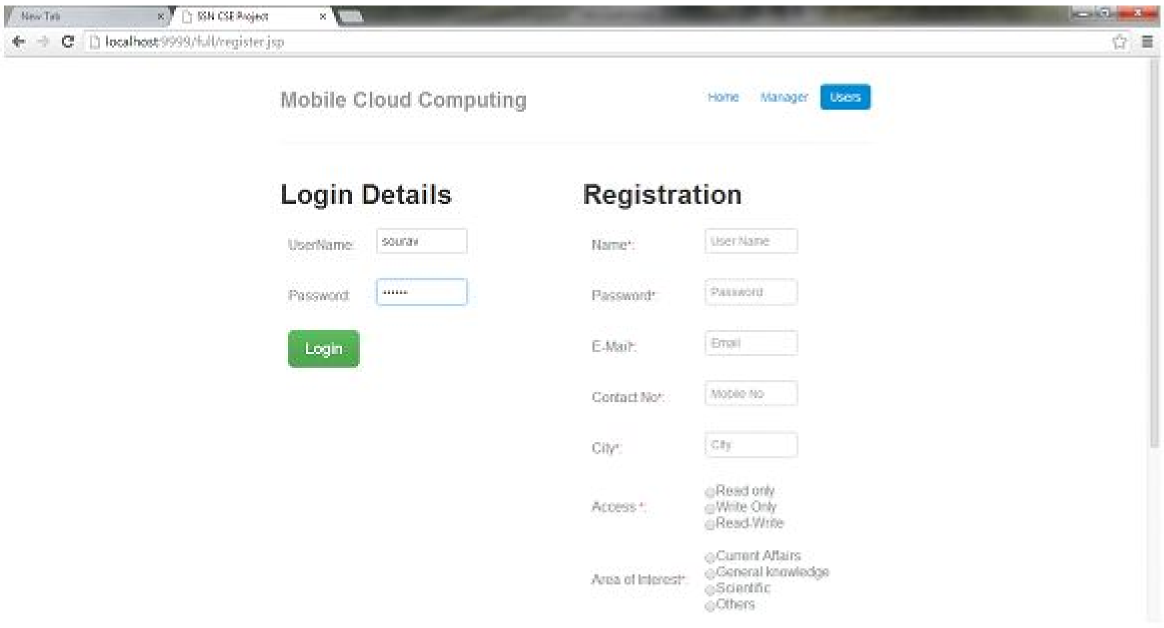


MySQL-Front - localhost - hybrid.user_registeration (Data Browser)

UserName	password	email	mobile	access	interest	Status	symetrickey
messi	messi	messi@gmail.com	9898989898	Read	Current Affairs	allow	65,66,67,68,69,70,7
xavi	xavi	xavi@gmail.com	9898989898	Write	General Knowledge	allow	65,66,67,68,69,70,7
anni	anni	anni@mail.com	9898989898	Read	General Knowledge	allow	65,66,67,68,69,70,7
annath	annath	aana@mail.co	9898989898	Write	Current Affairs	allow	65,66,67,68,69,70,7
rahul	rahul	rahul@mail.com	9884032060	Read	Others	allow	65,66,67,68,69,70,7
rams	rams12	ram@tcom.com	1111116589	Read	Others	allow	65,66,67,68,69,70,7
prashant	prashant	pgulesta1@gmail.con	9962531163	Write	General Knowledge	allow	65,66,67,68,69,70,7
hello	hello	hello@gmail.com	9898989898	Read	Scientific	allow	65,66,67,68,69,70,7
sachin	sachin	sach@gmail.com	9840622222	Write	Scientific	allow	65,66,67,68,69,70,7
sourav	sourav	sourav@gmail.com	9840622221	Read	General Knowledge	allow	65,66,67,68,69,70,7

UserName 6:0 30 Object(s) Connected since: 02:00

FIGURE 9.13: Database reflecting changes



Mobile Cloud Computing

Login Details

UserName: sourav

Password: ••••••

Registration

Name*: User Name

Password*: Password

E-Mail*: Email

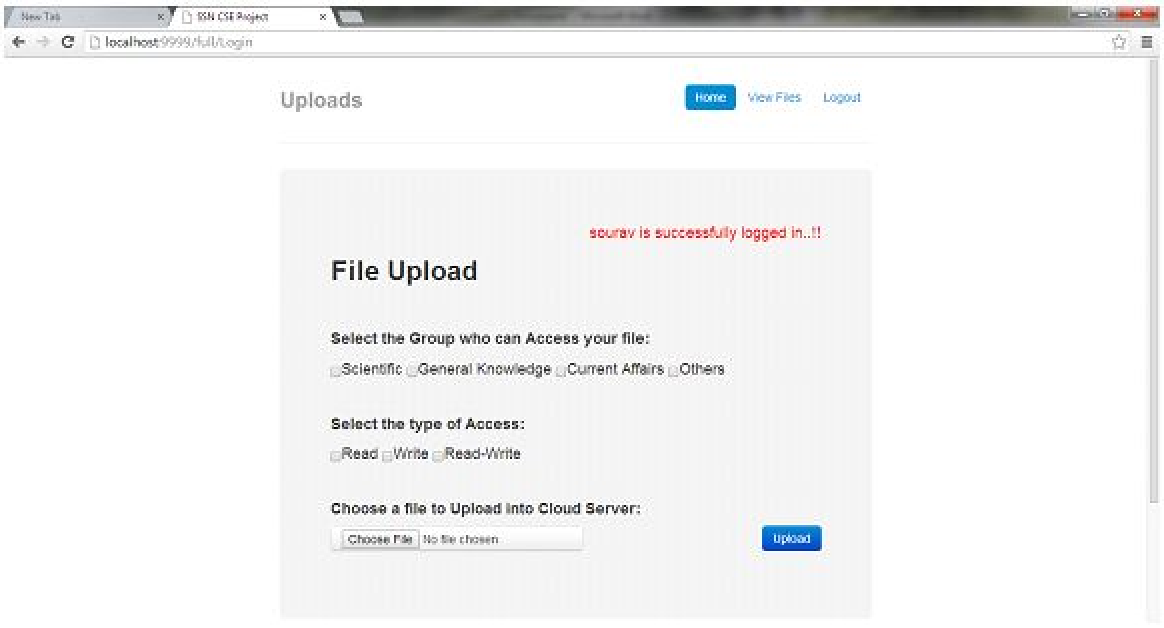
Contact No*: Mobile No

City*: City

Access *: Read only / Write Only / Read-Write

Area of Interest*: Current Affairs / General knowledge / Scientific / Others

FIGURE 9.14: User 1 login



Uploads

sourav is successfully logged in..!!

File Upload

Select the Group who can Access your file:

Scientific / General Knowledge / Current Affairs / Others

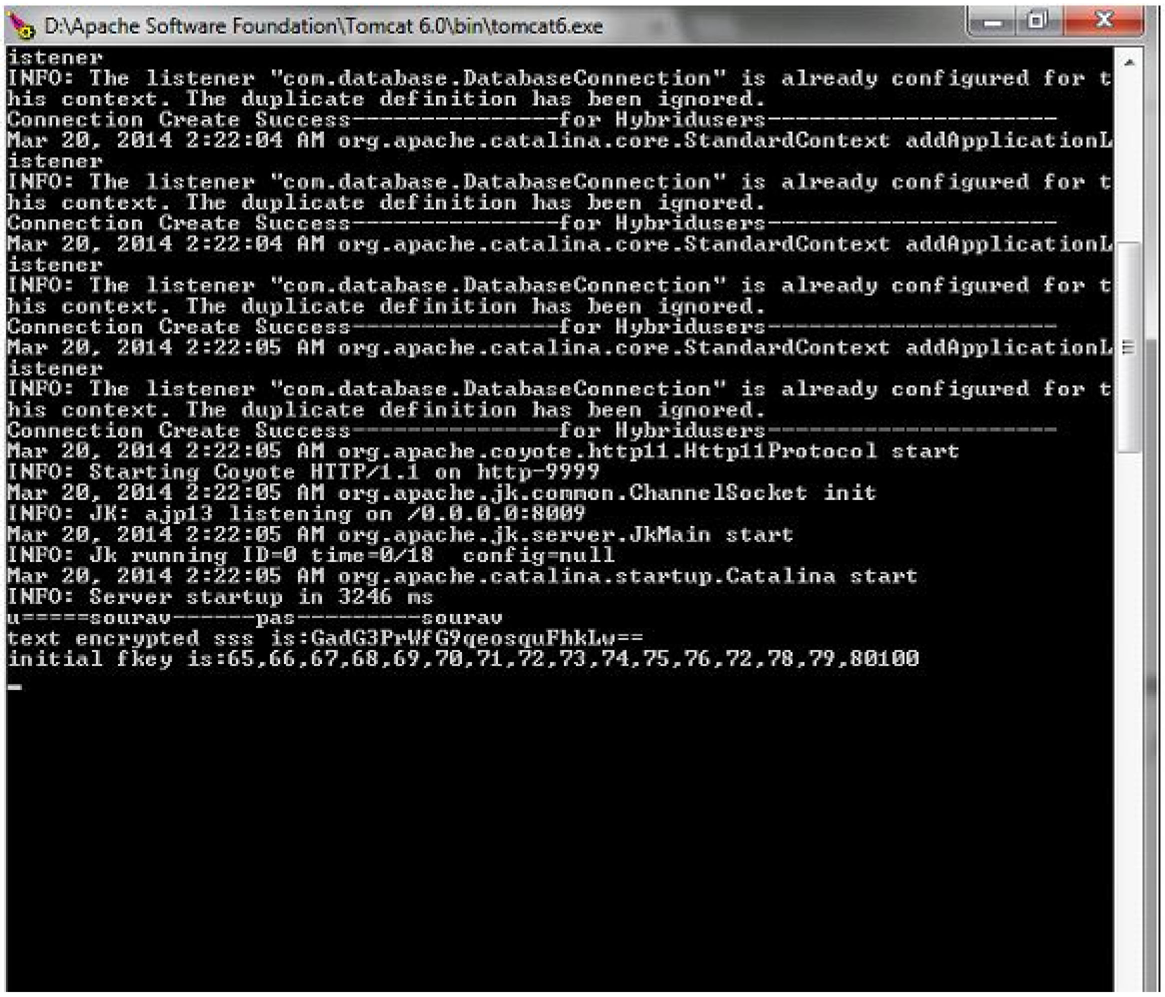
Select the type of Access:

Read / Write / Read-Write

Choose a file to Upload into Cloud Server:

No file chosen

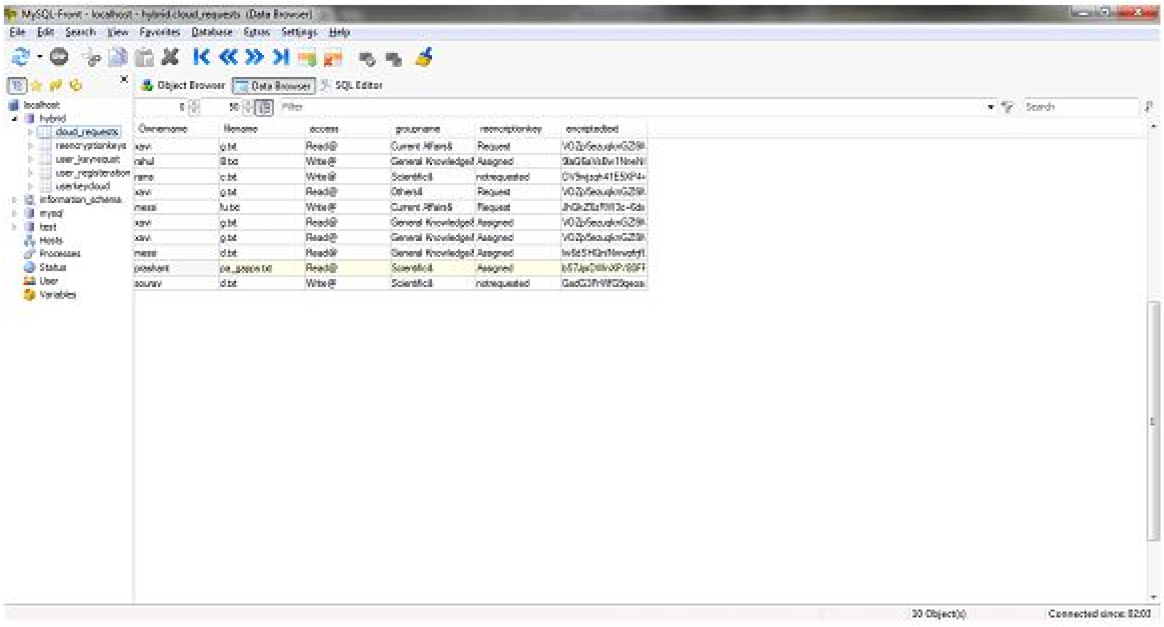
FIGURE 9.15: Upload page



D:\Apache Software Foundation\Tomcat 6.0\bin\tomcat6.exe

```
istener
INFO: The listener "com.database.DatabaseConnection" is already configured for t
his context. The duplicate definition has been ignored.
Connection Create Success---------------for Hybridusers---------------------
Mar 20, 2014 2:22:04 AM org.apache.catalina.core.StandardContext addApplicationL
istener
INFO: The listener "com.database.DatabaseConnection" is already configured for t
his context. The duplicate definition has been ignored.
Connection Create Success---------------for Hybridusers---------------------
Mar 20, 2014 2:22:04 AM org.apache.catalina.core.StandardContext addApplicationL
istener
INFO: The listener "com.database.DatabaseConnection" is already configured for t
his context. The duplicate definition has been ignored.
Connection Create Success---------------for Hybridusers---------------------
Mar 20, 2014 2:22:05 AM org.apache.catalina.core.StandardContext addApplicationL
istener
INFO: The listener "com.database.DatabaseConnection" is already configured for t
his context. The duplicate definition has been ignored.
Connection Create Success---------------for Hybridusers---------------------
Mar 20, 2014 2:22:05 AM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-9999
Mar 20, 2014 2:22:05 AM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Mar 20, 2014 2:22:05 AM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/18  config=null
Mar 20, 2014 2:22:05 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 3246 ms
u=====sourav-------pas----------sourav
text encrypted sss is:GadG3PrWfG9qeosquFhkLw==
initial fkey is:65,66,67,68,69,70,71,72,73,74,75,76,72,78,79,80100
```

FIGURE 9.16: Text contents on encryption



Ownername	filename	access	groupname	reencryptionkey	encryptedtext
xavi	g.txt	Read@	Current Affairs$	Request	VO2pSeouqknGZ9I
rahul	B.txt	Write@	General Knowledge$	Assigned	3taG6aVsBw1NneN
rama	c.txt	Write@	Scientific$	notrequested	OV9sqoh41E5XP4+
xavi	g.txt	Read@	Others$	Request	VO2pSeouqknGZ9I
messi	fu.txt	Write@	Current Affairs$	Request	JhGkZ6zRW3c-6ds
xavi	g.txt	Read@	General Knowledge$	Assigned	VO2pSeouqknGZ9I
xavi	g.txt	Read@	General Knowledge$	Assigned	VO2pSeouqknGZ9I
messi	d.txt	Read@	General Knowledge$	Assigned	lw5d5HGmNmwafg1
prashant	pa_paapa.txt	Read@	Scientific$	Assigned	b57JesOWnXP/S0F
sourav	d.txt	Write@	Scientific$	notrequested	GadG3PrWfG0qeos

FIGURE 9.17: Storing encrypted file contents

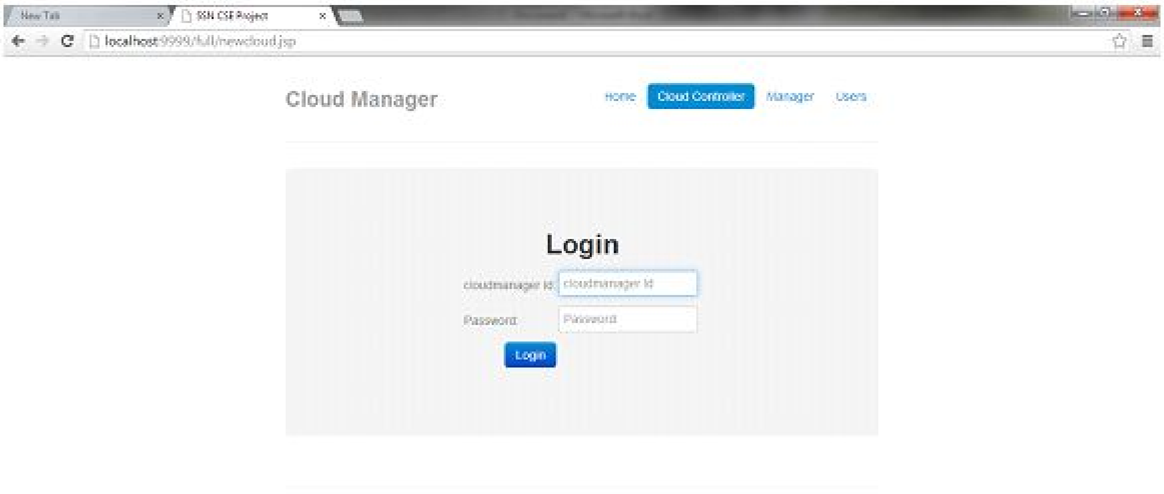
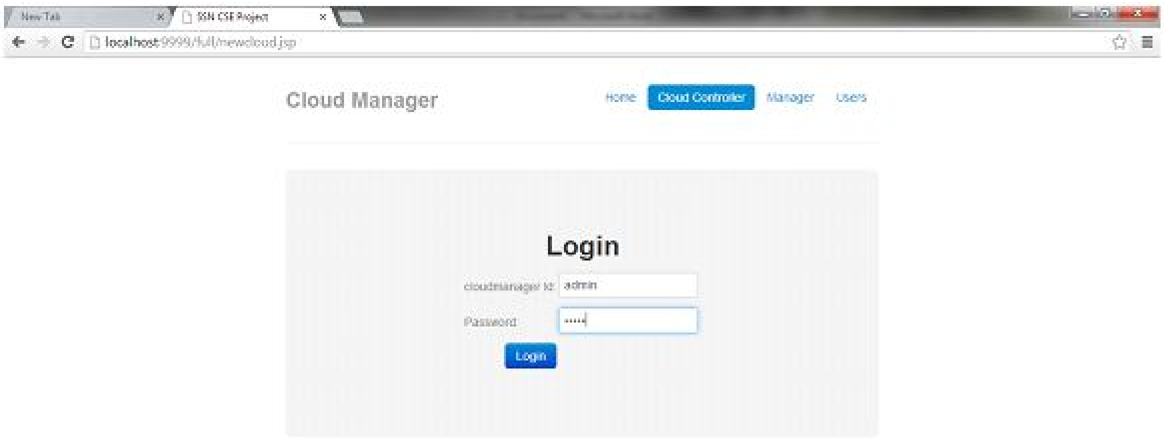


FIGURE 9.18: Cloud Controller



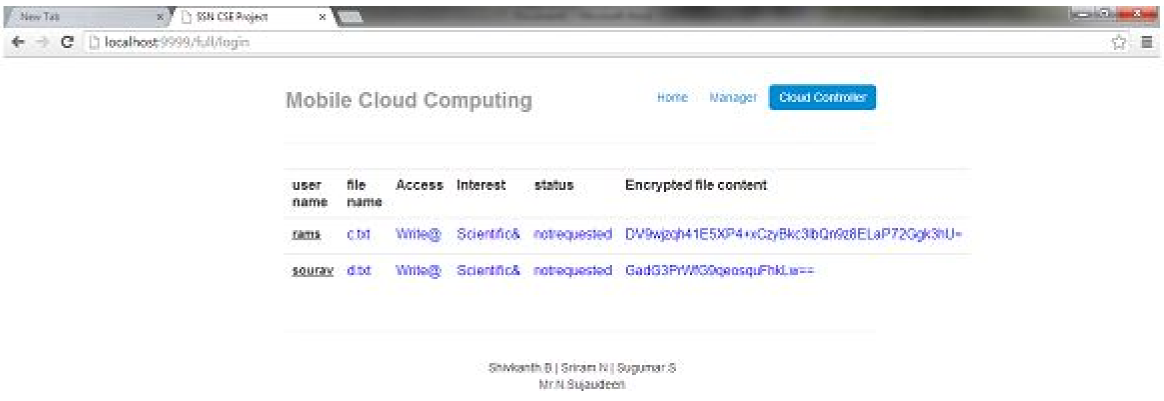
Cloud Manager

Login

cloudmanager id: admin

Password: •••••

FIGURE 9.19: Cloud Controller login

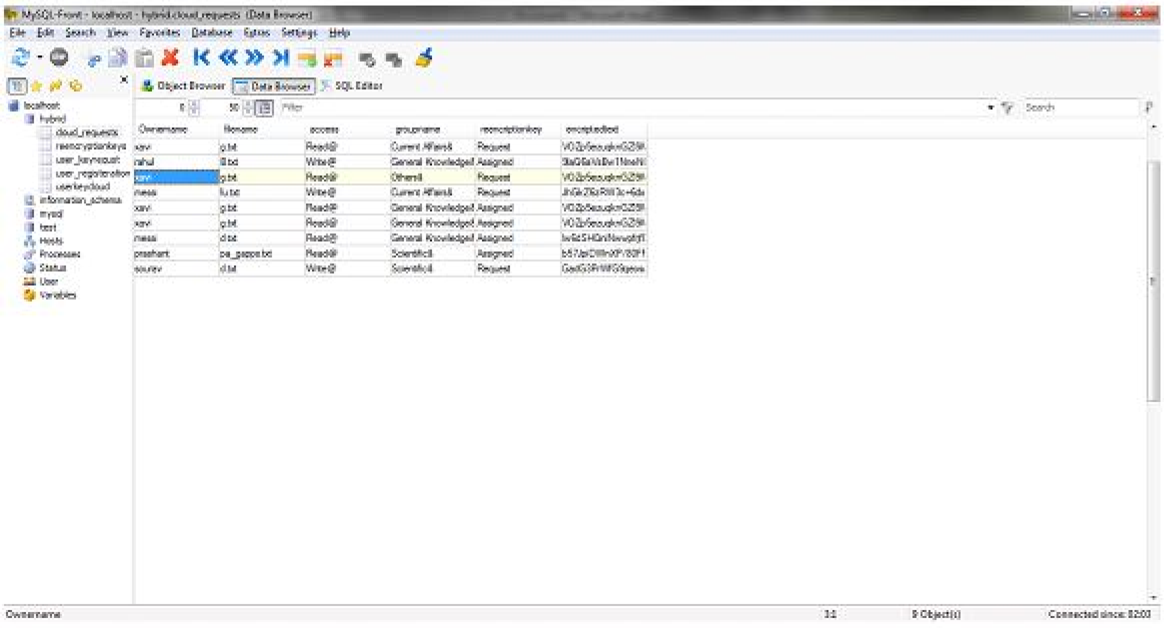


Mobile Cloud Computing

user name	file name	Access	Interest	status	Encrypted file content
rams	c.txt	Write@	Scientific&	notrequested	DV9wjzqh41E5XP4+xCzyBkc3lbQn9z8ELaP72Ggk3nU=
sourav	d.txt	Write@	Scientific&	notrequested	GadG3PYWG0qeosquFhkLe==

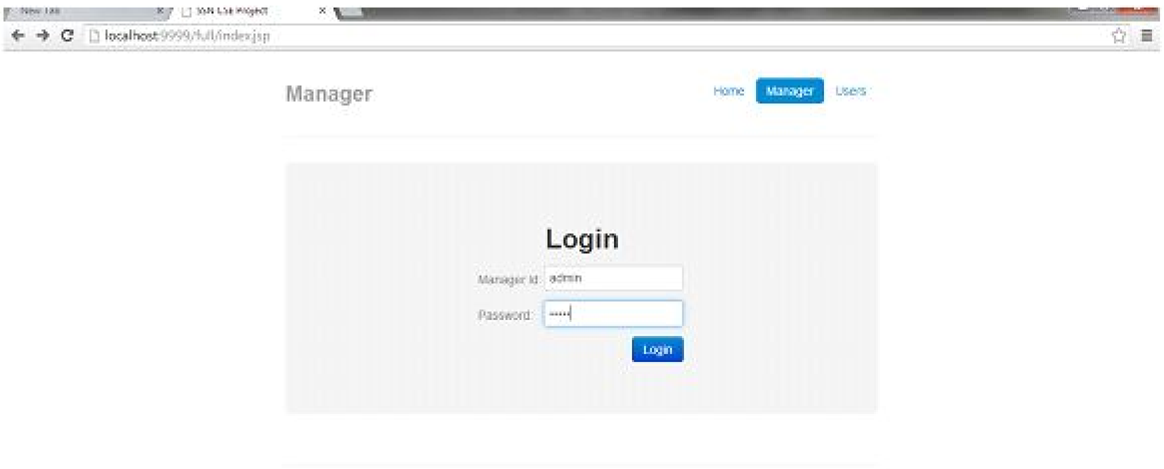
Shivkanth B | Sriram N | Sugumar S
Mr.N.Sujaudeen

FIGURE 9.20: On key request to Manager



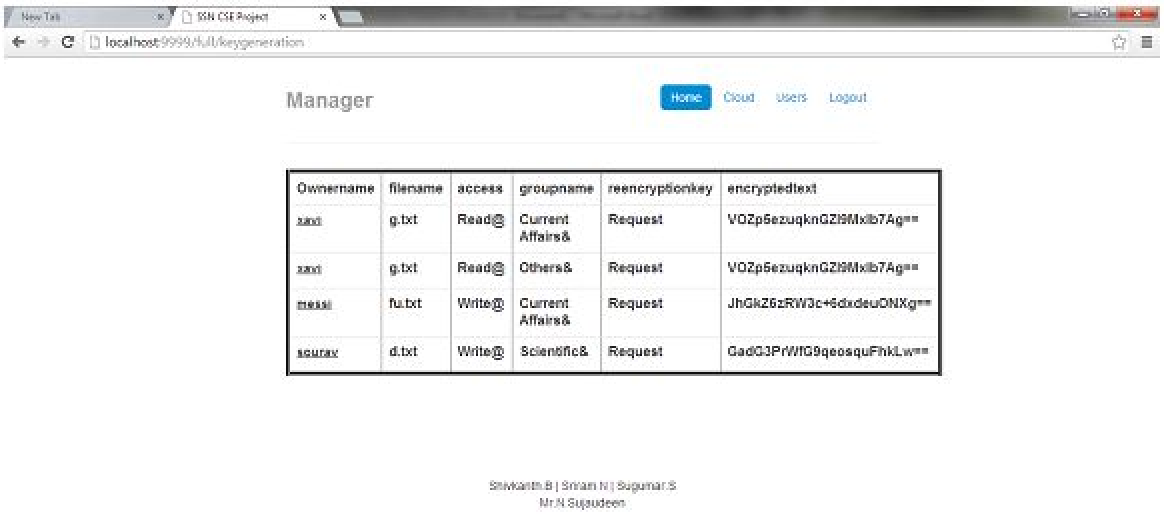
Ownername	filename	access	groupname	reencryptionkey	encryptedtext
xavi	g.txt	Read@	Current Affairs&	Request	VO2b5ezuqkrGZ9I
rahul	8.txt	Write@	General Knowledge&	Assigned	9aG6aVkBw1NneN
xavi	g.txt	Read@	Others&	Request	VO2b5ezuqkrGZ9I
messi	fu.txt	Write@	Current Affairs&	Request	JhGk2f6zRW3c+6dc
xavi	g.txt	Read@	General Knowledge&	Assigned	VO2b5ezuqkrGZ9I
xavi	g.txt	Read@	General Knowledge&	Assigned	VO2b5ezuqkrGZ9I
messi	d.txt	Read@	General Knowledge&	Assigned	lw6d5HGnNwvqfqf1
prashant	pa_paper.txt	Read@	Scientific&	Assigned	b57JeDWnXP/80F1
sourav	d.txt	Write@	Scientific&	Request	GadG3RrWfG9pew

FIGURE 9.21: Database reflecting changes



Manager Id: admin
Password: •••••

FIGURE 9.22: Manager login



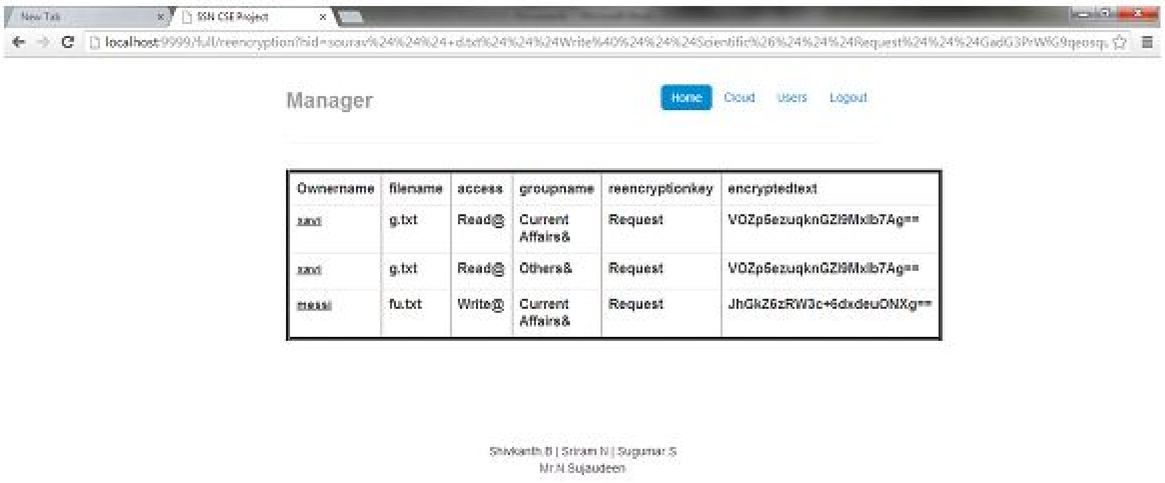
localhost:9999/full/keygeneration

Manager

Ownername	filename	access	groupname	reencryptionkey	encryptedtext
xavi	g.txt	Read@	Current Affairs&	Request	VOZp5ezuqknGZl9Mxlb7Ag==
xavi	g.txt	Read@	Others&	Request	VOZp5ezuqknGZl9Mxlb7Ag==
messi	fu.txt	Write@	Current Affairs&	Request	JhGkZ6zRW3c+6dxdeuONXg==
sourav	d.txt	Write@	Scientific&	Request	GadG3PrWfG9qeosquFhkLw==

Shivkanth.B | Sriram N | Sugumar.S
Mr.N Sujaudeen

FIGURE 9.23: Manager on key generation for re-encryption



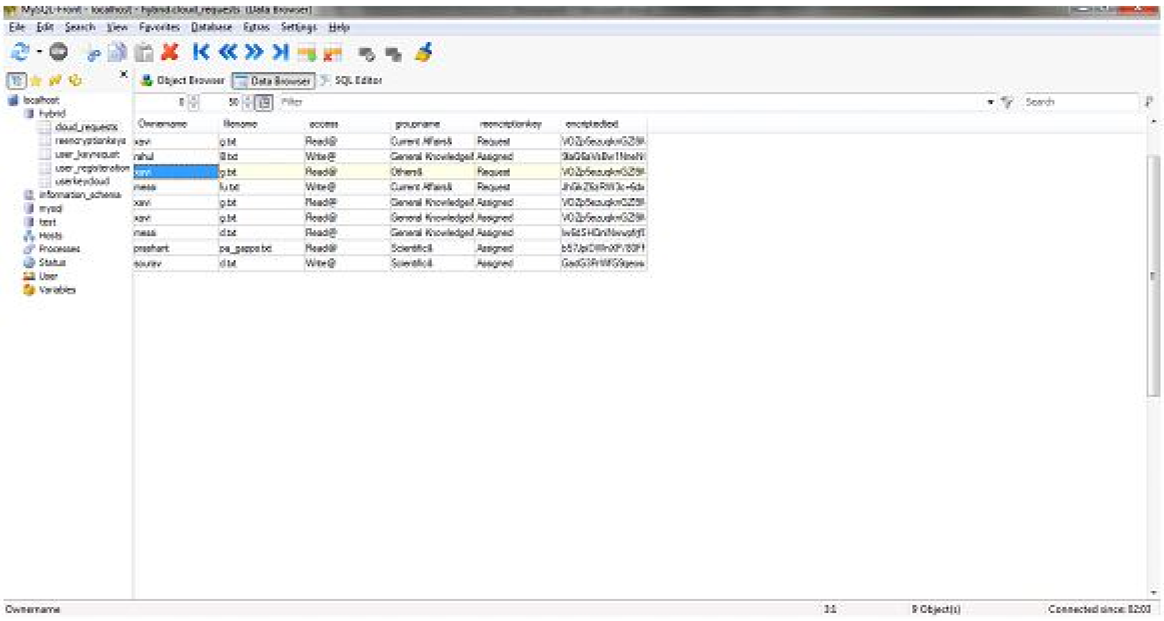
localhost:9999/full/reencryption?hid=sourav%24%24%24+d.txt%24%24%24Write%40%24%24%24Scientific%26%24%24%24Request%24%24%24GadG3PrWfG9qeosq

Manager

Ownername	filename	access	groupname	reencryptionkey	encryptedtext
xavi	g.txt	Read@	Current Affairs&	Request	VOZp5ezuqknGZl9Mxlb7Ag==
xavi	g.txt	Read@	Others&	Request	VOZp5ezuqknGZl9Mxlb7Ag==
messi	fu.txt	Write@	Current Affairs&	Request	JhGkZ6zRW3c+6dxdeuONXg==

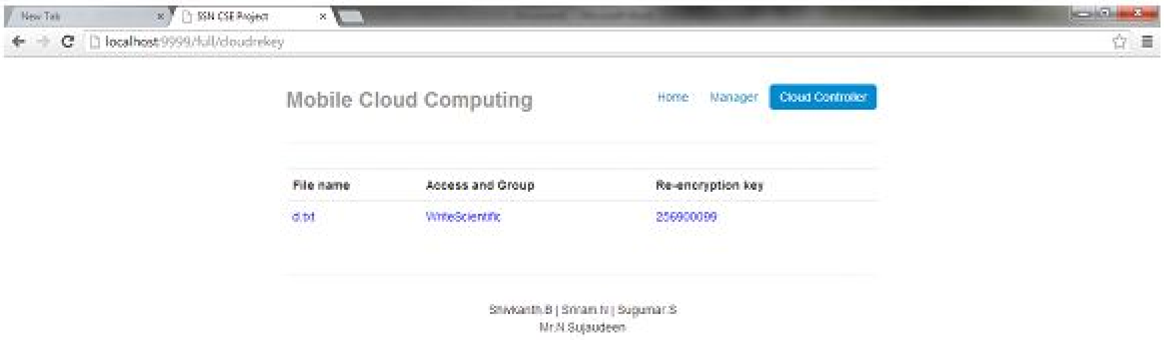
Shivkanth.B | Sriram N | Sugumar.S
Mr.N Sujaudeen

FIGURE 9.24: Manager key generation notification



Ownername	filename	access	groupname	reencryptionkey	encryptedtext
xavi	g.txt	Read@	Current Affairs&	Request	VOZpSeuqkmGZ9I
rahul	8.txt	Write@	General Knowledge&	Assigned	9aG6aVsBw1NneN
xavi	g.txt	Read@	Others&	Request	VOZpSeuqkmGZ9I
meea	fu.txt	Write@	Current Affairs&	Request	JhGkZ6qRW3c+6dx
xavi	g.txt	Read@	General Knowledge&	Assigned	VOZpSeuqkmGZ9I
xavi	g.txt	Read@	General Knowledge&	Assigned	VOZpSeuqkmGZ9I
meea	d.txt	Read@	General Knowledge&	Assigned	lw8dSHQnNewofjfI
prashant	pa_gappa.txt	Read@	Scientific&	Assigned	b57JpiOWmXP/80F1
sourav	d.txt	Write@	Scientific&	Assigned	GadG3FrWfG9geea

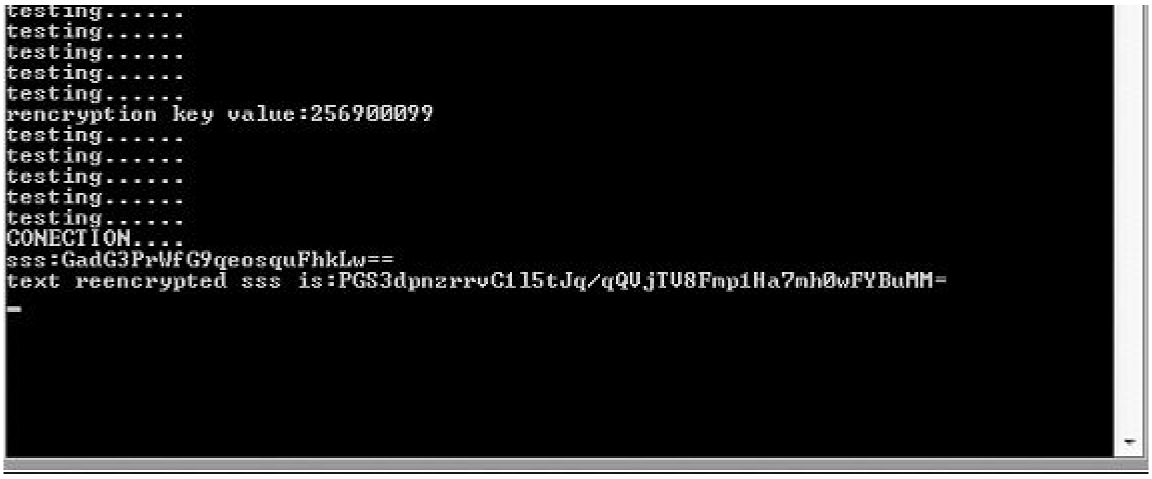
FIGURE 9.25: Re-encryption key assigned



File name	Access and Group	Re-encryption key
d.txt	WriteScientific	256900099

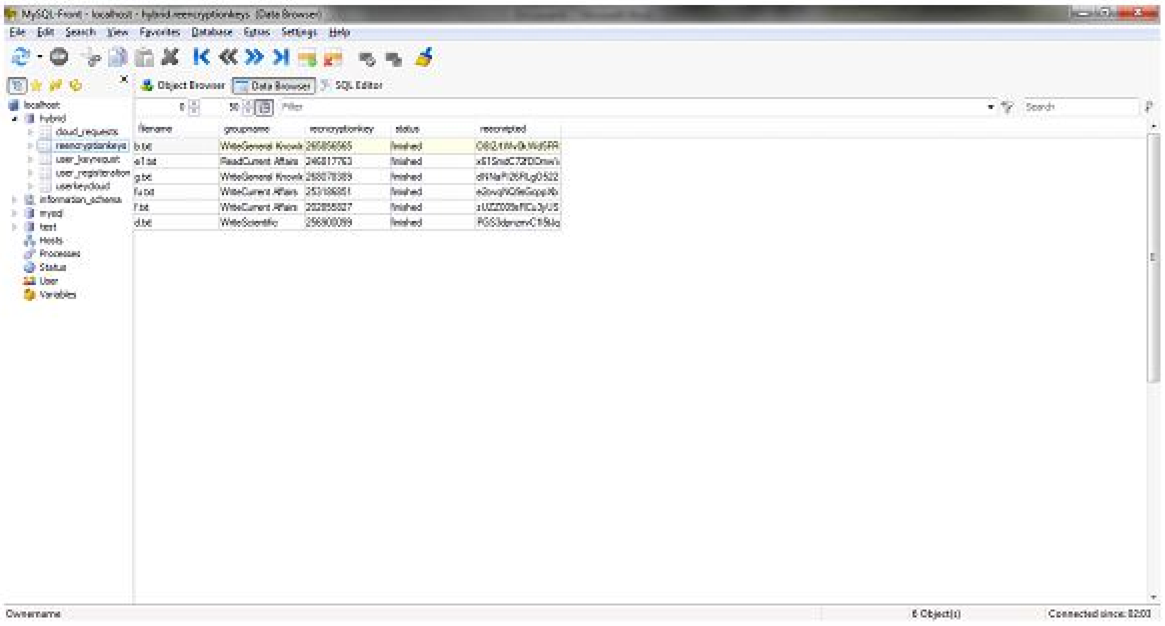
Shivkanth.B | Sriram N | Sugumar S
Mr.N Sujaudeen

FIGURE 9.26: Re-encryption key display



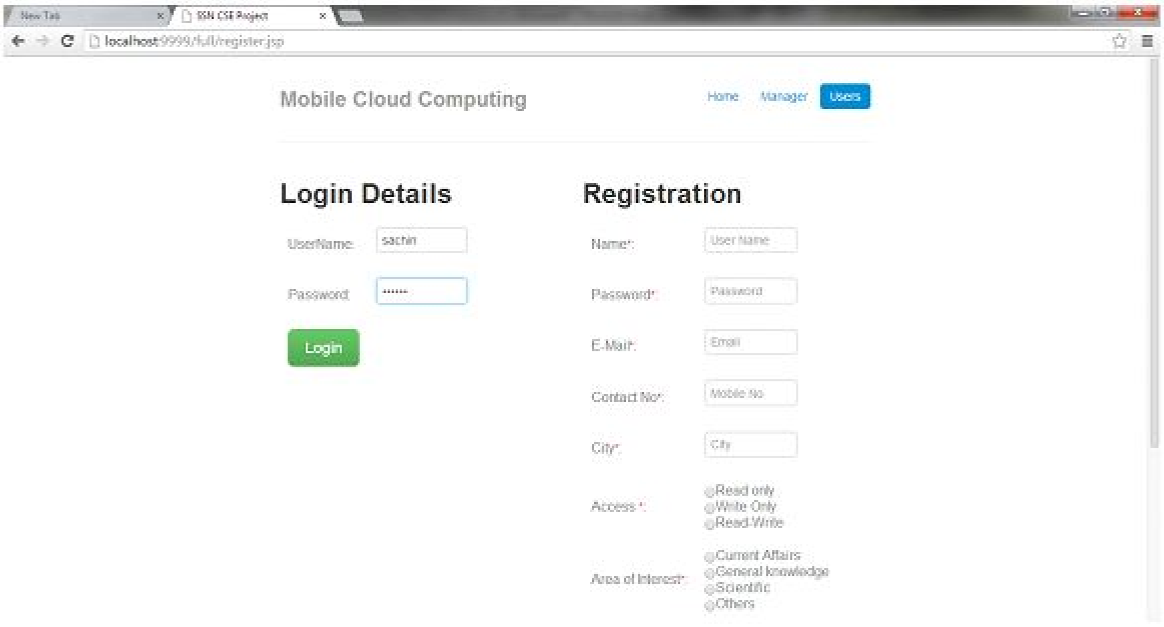
```
testing......
testing......
testing......
testing......
testing......
rencryption key value:256900099
testing......
testing......
testing......
testing......
testing......
CONECTION....
sss:GadG3PrWfG9qeosquFhkLw==
text reencrypted sss is:PGS3dpnzrrvC115tJq/qQVjTU8Fmp1Ha7mh0wFYBuMM=
_
```

FIGURE 9.27: Re-encrypted file contents on display



filename	groupname	reencryptionkey	status	reencrypted
b.txt	WriteGeneral Knowle	265056565	finished	C8i2.tWvBcWdSFR
a1.txt	ReadCurrent Affairs	246017763	finished	x61SnsC72fDDnw\
g.txt	WriteGeneral Knowle	260078389	finished	dNNaPi26RLg0522
fa.txt	WriteCurrent Affairs	253106851	finished	e2ovqNO9eGoppNb
f.txt	WriteCurrent Affairs	202055027	finished	zUZZ009eRCu3yUS
d.txt	WriteScientific	256900099	finished	PGS3dpnzrrvC115tJq

FIGURE 9.28: Database reflecting changes



Mobile Cloud Computing

Login Details

UserName: sachin

Password: ••••••

Login

Registration

Name*: User Name

Password*: Password

E-Mail*: Email

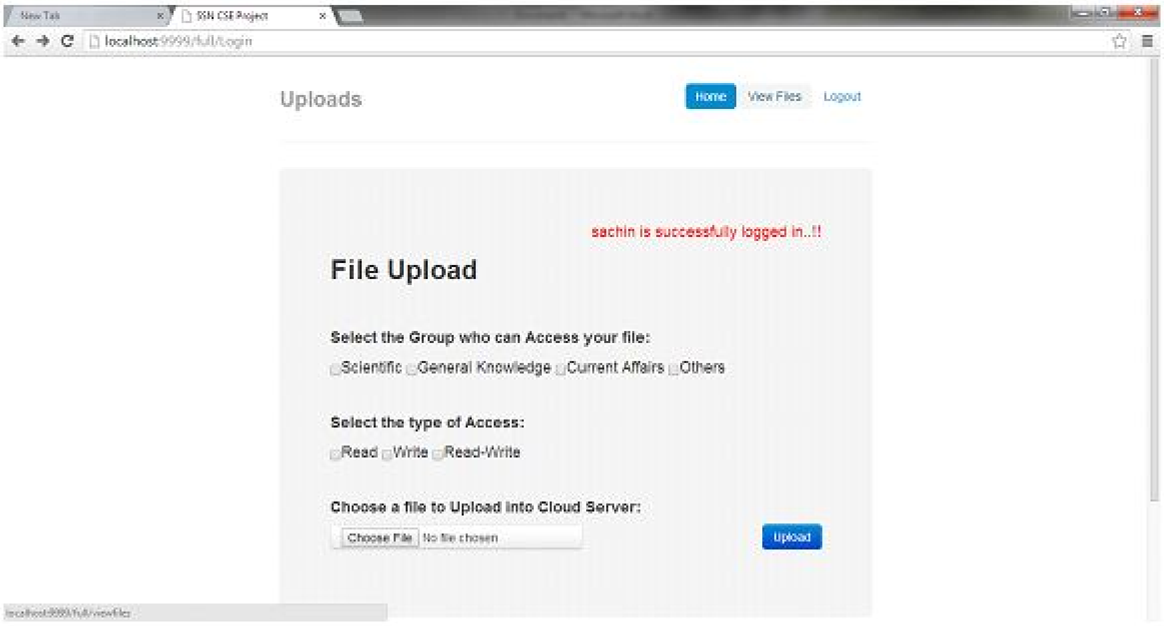
Contact No*: Mobile No

City*: City

Access *: Read only, Write Only, Read-Write

Area of Interest*: Current Affairs, General knowledge, Scientific, Others

FIGURE 9.29: User login



Uploads

sachin is successfully logged in..!!

File Upload

Select the Group who can Access your file:

Scientific General Knowledge Current Affairs Others

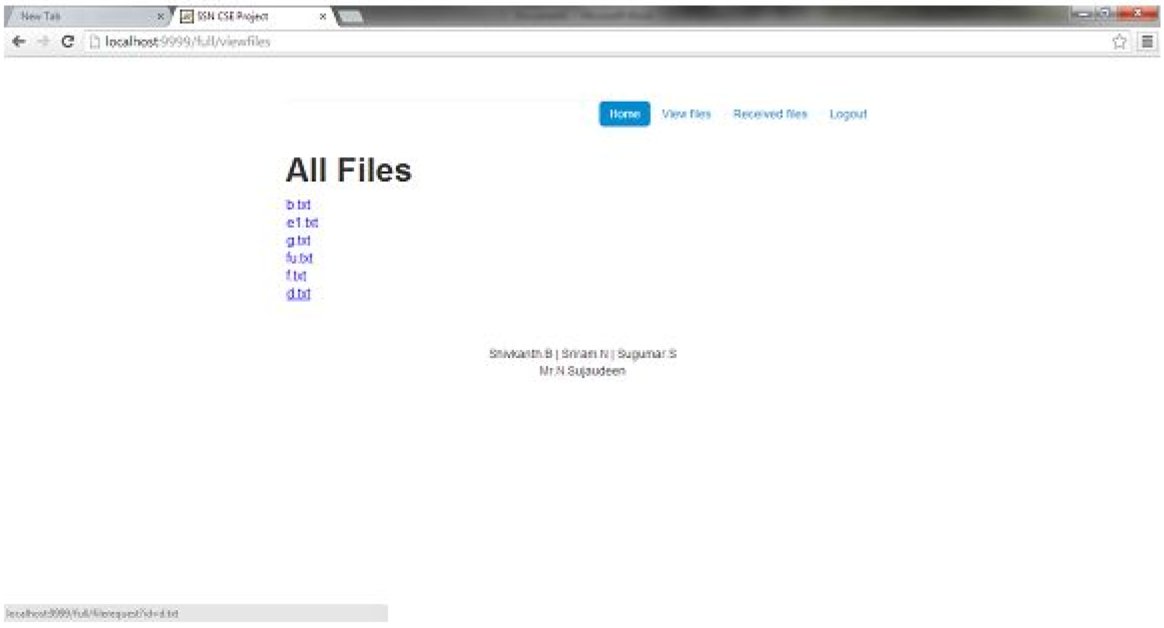
Select the type of Access:

Read Write Read-Write

Choose a file to Upload into Cloud Server:

Choose File No file chosen Upload

FIGURE 9.30: Login notification


New Tab | SSN CSE Project

localhost:9999/full/viewfiles

Home | View files | Received files | Logout

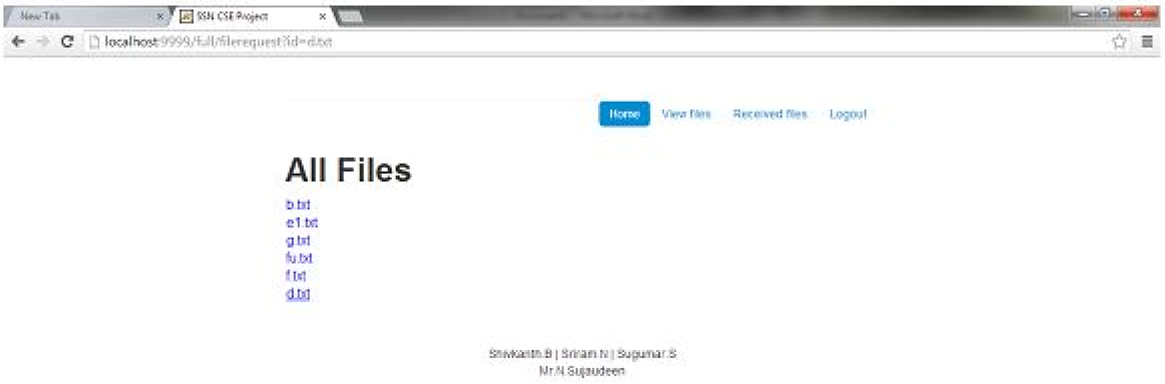
All Files

b.txt
e1.txt
g.txt
fu.txt
f.txt
d.txt

Shivkarth.B | Sriram N | Sugumar.S
Mr.N Sujaudeen

localhost:9999/full/filerequest?id=d.txt

FIGURE 9.31: Viewfiles


New Tab | SSN CSE Project

localhost:9999/full/filerequest?id=d.txt

Home | View files | Received files | Logout

All Files

b.txt
e1.txt
g.txt
fu.txt
f.txt
d.txt

Shivkarth.B | Sriram N | Sugumar.S
Mr.N Sujaudeen

FIGURE 9.32: User on key request for decryption

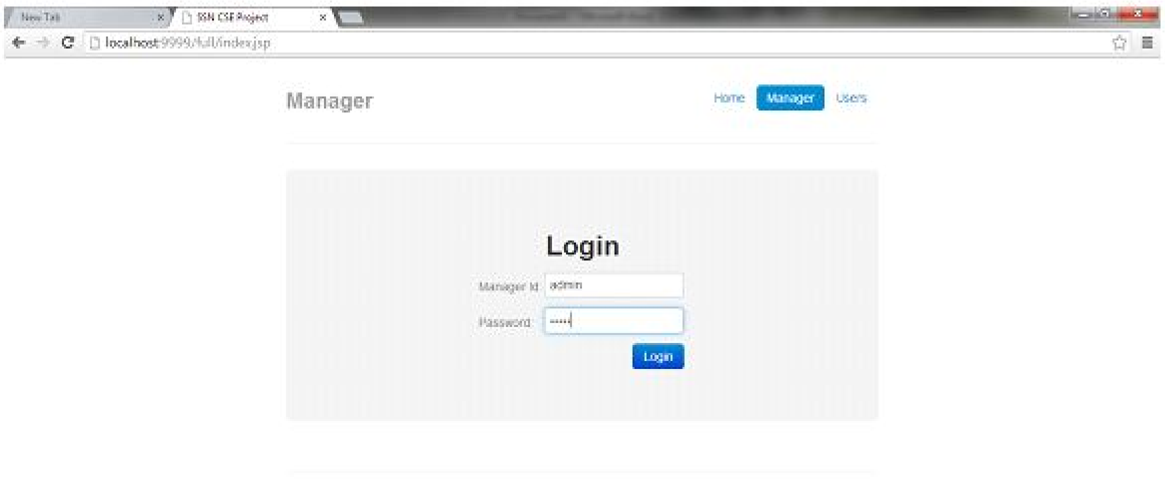
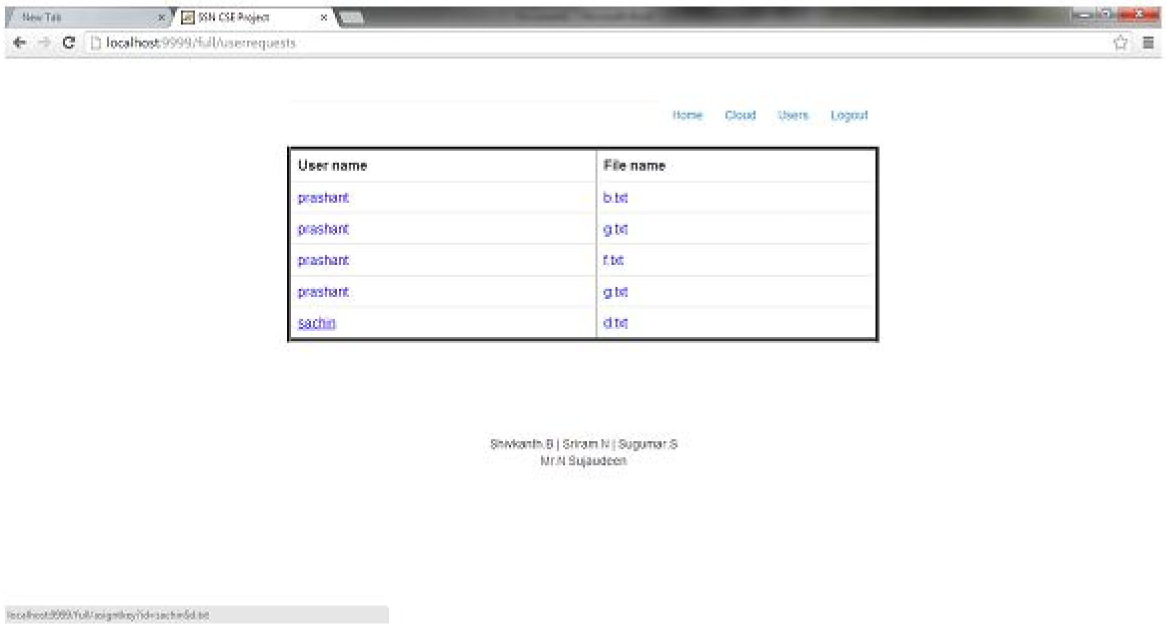


FIGURE 9.33: Manager login page

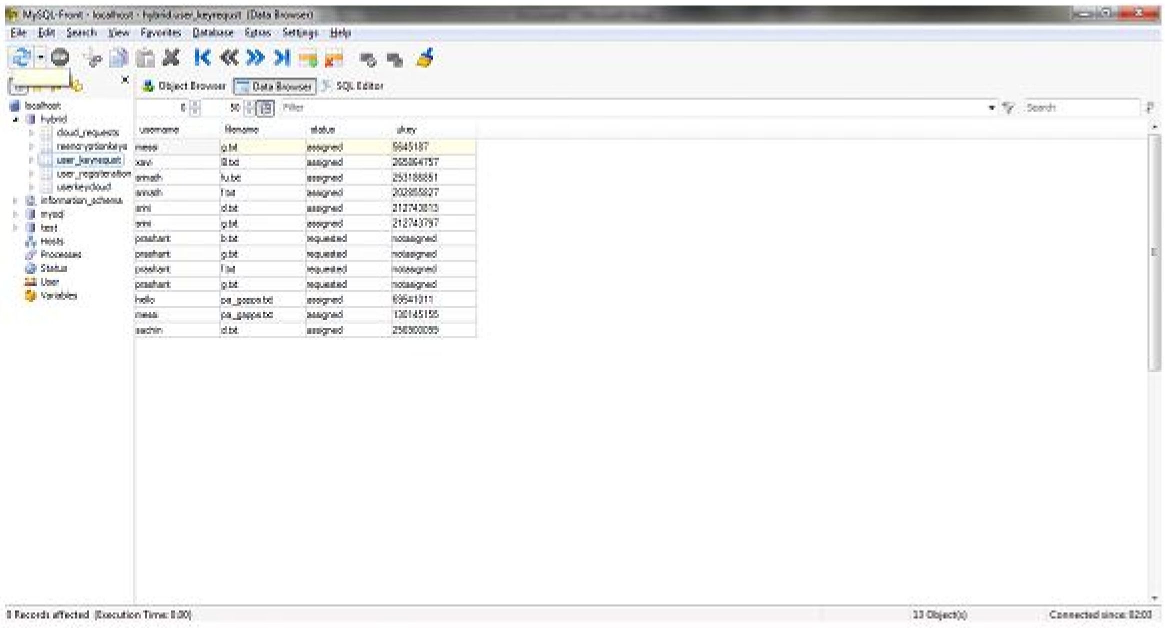


User name	File name
prashant	b.txt
prashant	g.txt
prashant	f.txt
prashant	g.txt
sachin	d.txt

Shivkanth.B | Sriram.N | Sugumar.S
Mr.N Sujaudeen

localhost:9999/full/assignkey?id=sachin&d.txt

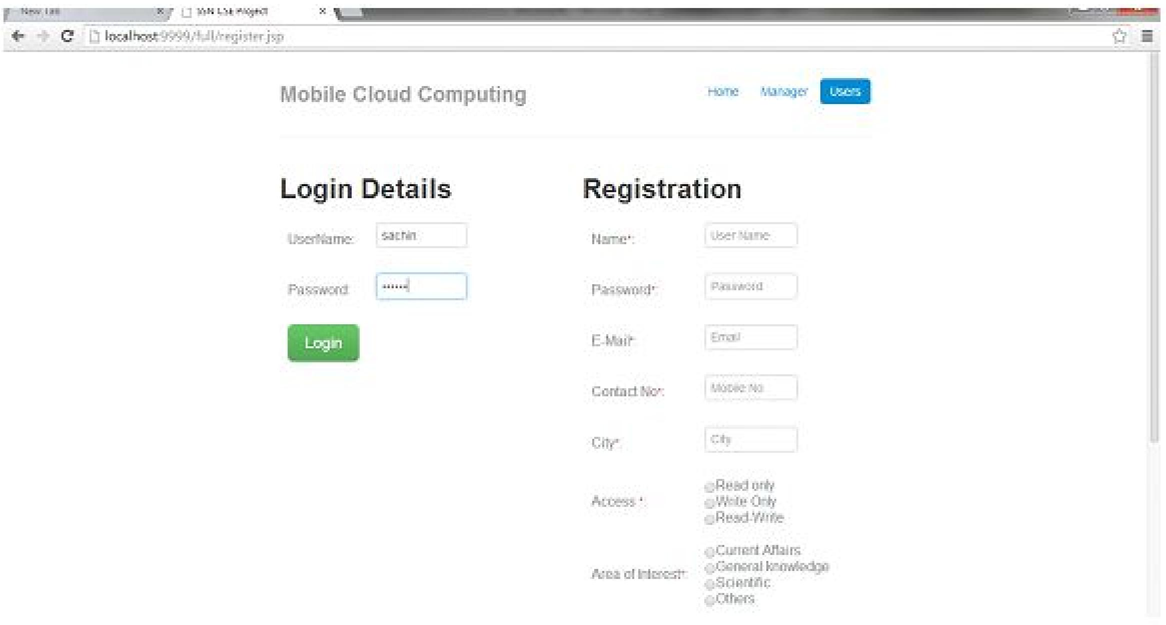
FIGURE 9.34: Manager granting user access based on attributes



username	filename	status	ukey
messi	g.txt	assigned	5645187
xavi	8.txt	assigned	265064757
amath	fu.txt	assigned	253188881
amath	f.txt	assigned	202855827
ami	d.txt	assigned	212743013
ami	g.txt	assigned	212743797
prashant	b.txt	requested	notassigned
prashant	g.txt	requested	notassigned
prashant	f.txt	requested	notassigned
prashant	g.txt	requested	notassigned
hello	pa_gappa.txt	assigned	69641011
messi	pa_gappa.txt	assigned	120145155
sachin	d.txt	assigned	258300095

0 Records affected (Execution Time: 0.00) 13 Object(s) Connected since: 02:03

FIGURE 9.35: Database reflecting changes



Login Details

UserName: sachin

Password: ••••••

Login

Registration

Name*: User Name

Password*: Password

E-Mail*: Email

Contact No*: Mobile No

City*: City

Access *: ◯Read only ◯Write Only ◯Read-Write

Area of Interest*: ◯Current Affairs ◯General knowledge ◯Scientific ◯Others

FIGURE 9.36: User login page

CHAPTER 10

Conclusion

The proposed protocol is similar in overall performance to the original ciphertext-policy attribute encryption idea, while significantly lessening the computational and traffic burden on the mobile data owner in a system where data updates and encryption activities are frequent and dominant. Thus, the proposal is useful for securing mobile cloud computing with very large user populations. A key management system has been proposed for secure data outsourcing applications, whereby attribute-based encryption effectively permits authorized users to access secure content in the cloud based on the satisfaction of an attribute based policy. The scheme has been modified so that a data owner and a trusted authority co-operate in the key generation and encryption processes such that computationally-intensive cryptographic operations and requests are minimized for the data owner. The above protocol could be expanded for applications with greater data values such as media files, streaming, money based transactions, client-server communication.

REFERENCES

1. Fahad Bin Nafey and OBV Ramanaiah. (2008) *A Study on Rijndael Algorithm for providing Condentiality to Mobile Devices,*in Region 10 IEEE Conference in TENCON

2. G.Ateniese, K. Fu, M. Green, and S. Hohenberger. (2006) *Improved proxyre-encryption schemes with applications to secure distributed storage,* ACM Transactions of Information and System Security, vol. 9, pp. 130.

3. G.Zhao, C. Rong, J. Li, F. Zhang, and Y. Tang. (2010) *Trusted data sharing over untrusted cloud storage providers,* in Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science, ser. CLOUDCOM 10. Washington, DC, USA: IEEE Computer Society.pp. 97103.

4. P. K. Tysowski and M. A. Hasan. (2011) *Towards Secure Communication for highly Scalable Mobile Applications in Cloud Computing Systems,* Centre for Applied Cryptographic Research (CACR), University of Waterloo, Tech. Rep. 33.

5. Foster, C., Uchitel, C., Magee, J. and Kramer J. (2003) *Model-based verication of web service compositions,* In proc. of ASE03, pp. 152-163.

6. Q. Liu, G. Wang, and J. Wu. (2012) *Clock-based proxy re-encryption scheme in unreliable clouds*, in Parallel Processing Workshops (ICPPW) 41st International Conference on, sept. 2012, pp. 304 305 .

7. S. Jahid, P. Mittal, and N. Borisov. (2011) *EASiER: encryption-based access control in social networks with efficient revocation*, in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS 11 New York, NY, USA: ACM, ,pp. 411415