

ANONYMOUS DATA SHARING SCHEME IN PUBLIC CLOUD AND ITS APPLICATION IN E-HEALTH RECORD

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

by

KALYANAPU MANINDRANATH(Reg.No.38110227)

JONNALA NIKHILSAI REDDY(Reg. No.38110369)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU**

March 2022



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

1.1.1 (DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119

www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KALYANAPU MANINDRANATH (Reg.No.38110227)** and **JONNALA NIKHIL SAI REDDY (Reg.No.38110369)** who carried out the project entitled "**ANONYMOUS DATA SHARING SCHEME IN PUBLIC CLOUD AND ITS APPLICATION IN E-HEALTH RECORD**" under my supervision from **August 2021** to **March 2022**.

Internal Guide

Dr. Mercy Paul Selvan, M.E., Ph D.,

Head of the Department

Dr. LAKSHMAN M.E., Ph. D

Submitted for Viva voce Examination held on _____

Internal Examiner External Examiner

DECLARATION

I, KALYANAPUMANINDRANATH and JONNALA NIKHIL SAI REDDY

here by declare that the Professional Training Report entitled “**ANONYMOUS DATA SHARING SCHEME IN PUBLIC CLOUD AND ITS APPLICATION IN E-HEALTH RECORD**” done by me under the guidance of **Dr, MERCY PAUL SELVAN, M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D., Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D.,** and **Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.MERCY PAUL SELVAN, M.E., Ph.D.,**for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Your health care provider may be moving from paper records to electronic health records or may be using EHRs already. EHRs allow providers to use information more effectively to improve the quality and efficiency of your care, but EHRs will not change the privacy protections or security safeguards that apply to your health information. This project focuses on developing secure cloud framework for evolving and accessing trusted computing services in all levels of public cloud deployment model. Thus, eliminates both internal and external security threats. These results in achieving data confidentiality, data integrity, authentication and authorization, eliminating both active and passive attacks from cloud network environment. To develop a secure cloud framework for accessing trusted computing and storage services in all levels of public cloud deployment model.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No
	ABSTRACT	V
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 OVERVIEW	1
2	LITERATURE SURVEY	3
	2.1 ANALYSIS OF THE LITERATURE	3
	2.2 LITERARY REVIEWS	10
3	SYSTEM ANALYSIS	
	3.1 PROJECT SCOPE	11
	3.2 OBJECTIVE	11
	3.3 EXISTING SYSTEM	11
	3.4 PROPOSED SYSTEM	11
	3.5 HARD REQUIREMENT	12

	3.6 SOFTWARE REQUIREMENT	12
	3.7 SOFTWARE DESCRIPTION	12
	3.7.1 JAVA	12
	3.7.2 MYSQL	18
	3.7.3 SYSTEM DESGIN AND TESTING	22
4	METHODOLOGY AND ALGORITHM USED	27
	4.1 SYSTEM ARCHITECTURE	27
	4.2 PROPOSED ALGORITHMS	27
	4.2.1 SHA ALGORITHM	27
	4.2.2 ADVANCE ENCRYPTION	28
	4.2.3 DATA ENCRYPTION STANDARD	28
	4.2.4 CHUNKING TECHNIQUES	29
	4.3 MODULES	30
	4.3.1 LOGIN MODULE	30

	4.3.2 REGISTRATION MODULE	30
	4.3.3 CREATION STORAGE AND INSTANCES	31
	4.3.4 DATA PROTECTION	31
	4.3.5 DATA RECOVERY MODULE	32
5	RESULTS AND DISCUSSION	32
6	SUMMARY AND CONCLUSION	41
	6.1 CONCLUSION	42
	REFERENCES	43
	APPENDIX	43
	SCREENSHOTS	44
	SOURCE CODE	46
	PLAGIARISM REPORT	50

LIST OF FIGURES

FIGNO	FIGURES NAMES	PAGE NO
5.1	HOME PAGE	32
5.2	LOGIN PAGE	32
5.3	RESISTRATION PAGE	33
5.4	FILE UPLOAD	33
5.5	TOKEN REQUEST	34
5.6	PRIVATE CLOUD	34

CHAPTER -1

INTRODUCTION

1.1 OVERVIEW

With the explosive growth of data, it is a heavy burden for users to store the sheer amount of data locally. Therefore, more and more organizations and individuals would like to store their data in the cloud. However, the data stored in the cloud might be corrupted or lost due to the inevitable software bugs, hardware faults and human errors in the cloud. In order to verify whether the data is stored correctly in the cloud, many remote data integrity auditing schemes have been proposed. In remote data integrity auditing schemes, the data owner firstly needs to generate signatures for data blocks before uploading them to the cloud. These signatures are used to prove the cloud truly possesses these data blocks in the phase of integrity auditing. And then the data owner uploads these data blocks along with their corresponding signatures to the cloud. The data stored in the cloud is often shared across multiple users in manycloud storage applications, such as Google Drive, Dropbox and iCloud. Data sharing as one of the most common features in cloud storage, allows a number of users to share their data with others. However, these shared data stored in the cloud might contain some sensitive information. For instance, the Electronic Health Records stored and shared in the cloud usually contain patients' sensitive information (patient's name, telephone number and ID number, etc.) and the hospital's sensitive information (hospital's name, etc.). If these EHRs are directly uploaded to the cloud to be shared for research purposes, the sensitive information of patient and hospital will be inevitably exposed to the cloud and the researchers. Besides, the integrity of the EHRs needs to be guaranteed due to the existence of human errors and software/hardware failures in the cloud. Therefore, it is important to accomplish remote data integrity auditing on the condition that the sensitive information of shared data is protected. A potential method of solving this problem is to encrypt the whole shared file

before sending it to the cloud, and then generate the signatures used to verify the integrity of this encrypted file, finally upload this encrypted file and its corresponding signatures to the cloud. This method can realize the sensitive information hiding since only the data owner can decrypt this file. However, it will make the whole shared file unable to be used by others. For example, encrypting the EHRs of infectious disease patients can protect the privacy of patient and hospital, but these encrypted EHRs cannot be effectively utilized by researchers any more. Distributing the decryption key to the researchers seems to be a possible solution to the above problem. However, it is infeasible to adopt this method in real scenarios due to the following reasons. Firstly, distributing decryption key needs secure channels, which is hard to be satisfied in some instances. Furthermore, it seems very difficult for a user to know which researchers will use his/her EHRs in the near future when he/she uploads the EHRs to the cloud. As a result, it is impractical to hide sensitive information by encrypting the whole shared file. Thus, how to realize data sharing with sensitive information hiding in remote data integrity auditing is very important and valuable. Unfortunately, this problem has remained unexplored in previous research.

CHAPTER -2

LITERATURE SURVEY

2.1 ANALYSIS OF THE LITERATURE

Literature survey is the main advance in programming improvement measure. Prior to building up the instrument it is important to decide the time factor, economy and friends strength. When these things are fulfilled, at that point the subsequent stage is to figure out which working framework and language can be utilized for building up the device. When the developers begin assembling the apparatus the software engineers need parcel of outer help. This help can be gotten from senior developers, from book or from sites. The major part of the project development sector considers and fully survey all the required needs for developing the project. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Prior to building the framework the above thought are considered for building up the proposed framework. The significant piece of the undertaking advancement area considers and completely survey all the necessary requirements for building up the venture. For each undertaking Literature survey is the main area in programming improvement measure. Prior to building up the instruments and the related planning it is important to decide and survey the time factor, asset prerequisite, labor, economy, and friends strength. When these things are fulfilled and completely surveyed, at that point the following stage is to decide about the product details in the separate framework, for example, what kind of working framework the venture would require and what are largely the important programming are expected to continue with the subsequent stage like building up the apparatuses, and the related activities. Here we have taken the general surveys of different creators and noted down the fundamental central issues with respect to their work. In this venture literature survey assumes a prevailing part in getting assets from different areas and all the connected points that are exceptionally valuable

under this segment. The most awesome aspect of this is the manner in which things get all together and encourages us to suite our work according to the current information.

2.2 LITERARY REVIEWS

TITLE 1:

Securely Outsourcing Attribute-Based Encryption with Checkability.

AUTHOR: Jin Li, Xinyi Huang, Jingwei Li, Xiaofeng Chen, and Yang Xiang

Attribute-Based Encryption (ABE) is a promising cryptographic primitive which significantly enhances the versatility of access control mechanisms. Due to the high expressiveness of ABE policies, the computational complexities of ABE key-issuing and decryption are getting prohibitively high. Despite that the existing Outsourced ABE solutions are able to offload some intensive computing tasks to a third party, the verifiability of results returned from the third party has yet to be addressed. Aiming at tackling the challenge above, we propose a new Secure Outsourced ABE system, which supports both secure outsourced key-issuing and decryption. Our new method offloads all access policy and attribute related operations in the key-issuing process or decryption to a Key Generation Service Provider (KGSP) and a Decryption Service Provider (DSP), respectively, leaving only a constant number of simple operations for the attribute authority and eligible users to perform locally. In addition, for the first time, we propose an outsourced ABE construction which provides checkability of the outsourced computation results in an efficient way. Extensive security and performance analysis show that the proposed schemes are proven secure and practical.

TITLE: 2 Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records

AUTHOR: Josh Benaloh, Melissa Chase, Eric Horvitz, and Kristin Lauter

We explore the challenge of preserving patients' privacy in electronic health record systems. We argue that security in such systems should be enforced via encryption as well as access control. Furthermore, we argue for approaches that enable patients to generate and store encryption keys, so that the patients' privacy is protected should the host data center be compromised. The standard argument against such an approach is that encryption would interfere with the functionality of the system. However, we show that we can build an efficient system that allows patients both to share partial access rights with others, and to perform searches over their records. We formalize the requirements of a Patient Controlled Encryption scheme, and give several instantiations, based on existing cryptographic primitives and protocols, each achieving a different set of properties.

TITLE 3: Cross-Domain Data Sharing in Distributed Electronic Health Record Systems

AUTHOR: Jinyuan Sun; Yuguang Fang

domain delegation scenario of interest.

Cross-organization or cross-domain cooperation takes place from

2 time to time in Electronic Health Record (EHR) system for necessary and high-quality patient treatment. Cautious design of delegation mechanism must be in place as a building block of cross-domain cooperation, since the cooperation inevitably involves exchanging and sharing relevant patient data that are considered highly private and confidential. The delegation mechanism grants permission to and restricts access rights of a cooperating partner. Patients are unwilling to accept the EHR system unless their health data are guaranteed proper use and disclosure, which cannot be easily achieved without cross-domain authentication and fine-grained access control. In addition, revocation of the delegated rights should be possible at any time during the cooperation. In this paper, we propose a secure EHR system, based on cryptographic constructions, to enable secure sharing of sensitive patient data during cooperation and preserve patient data privacy. Our EHR system further incorporates advanced mechanisms for fine-grained access control, and on-demand revocation, as enhancements to the basic

access control offered by the delegation mechanism, and the basic revocation mechanism, respectively. The proposed EHR system is demonstrated to fulfill objectives specific to the cross-domain delegation scenario of interest.

TITLE 4:Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data

AUTHOR:Ning Cao ; Cong Wang ; Ming Li ; Kui Ren ; Wenjing Lou

With the advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data have to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely sort the search results. In this paper, for the first time, we define and solve the challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). We establish a set of strict privacy requirements for such a secure cloud data utilization system. Among various multi-keyword semantics, we choose the efficient similarity measure of "coordinate matching," i.e., as many matches as possible, to capture the relevance of data documents to the search query. We further use "inner product similarity" to quantitatively evaluate such similarity measure. We first propose a basic idea for the MRSE based on secure inner product computation, and then give two significantly improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. To improve search experience of the data search service, we further extend these two schemes to support more search semantics. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given. Experiments on the real-world data set further show proposed schemes indeed introduce low overhead on computation and communication.

TITLE:5 Privacy-Preserving Cloud-Based Personal Health Record System Using Attribute-Based Encryption And Anonymous Multi-Receiveridentity-Based Encryption

AUTHOR:Changji Wang, Xilei Xu, Dongyuan Shi, Jian Fang

As an emerging patient-centric model of health information exchange, cloud-based per-sonal health record (CB-PHR) system holds great promise for empowering patients andensuring more efective delivery of health care. In this paper, we design a novel CB-PHRsystem. It allows PHR owners to securely store their health data on the semi-trustedcloud service providers, and to selectively share their health data with a wide rangeof PHR users. To reduce the key management complexity, we divide PHR users intotwo security domains named public domain and personal domain. PHR owners encrypttheir health data for the public domain using ciphertext-policy attribute-based encryp-tion scheme, while encrypt their health data for the personal domain using anonymousmulti-receiver identity-based encryption scheme. Only authorized users whose creden-tials satisfy the specified ciphertext-policy or whose identities belong to dedicated iden-tities can decrypt the encrypted health data. Extensive analytical and experimentalresults are presented which show that our CB-PHR system is secure, privacy-protected,scalable and efcient.

TITLE: 6 CDPS: A cryptographic data publishing system

AUTHOR:T. Li, Z. Liu, J. Li, C. Jia, K. Li

The traditional data publishing methods will remove the sensitive attributes and generate the abundant records to achieve the goal of privacy protection. In the big data environment, the requirement of utilizing data (e.g., data mining) become more and more various, which is beyond the scope of the traditional method. This paper provides a cryptographic data publishing system that preserves the data integrity (i.e., the original data structure is preserved) and achieves anonymity without deletion of any attribute or utilization of redundancy. The security analysis shows that our system is secure under our proposed security model.

TITLE 7:Identity-based encryption with outsourced revocation in cloud

computing

AUTHOR: J. Li, J. W. Li, X. Chen, C. Jia, W. Lou

Identity-Based Encryption (IBE) which simplifies the public key and certificate management at Public Key Infrastructure (PKI) is an important alternative to public key encryption. However, one of the main efficiency drawbacks of IBE is the overhead computation at Private Key Generator (PKG) during user revocation. Efficient revocation has been well studied in traditional PKI setting, but the cumbersome management of certificates is precisely the burden that IBE strives to alleviate. In this paper, aiming at tackling the critical issue of identity revocation, we introduce outsourcing computation into IBE for the first time and propose a revocable IBE scheme in the server-aided setting. Our scheme offloads most of the key generation related operations during key-issuing and key-update processes to a Key Update Cloud Service Provider, leaving only a constant number of simple operations for PKG and users to perform locally. This goal is achieved by utilizing a novel collusion-resistant technique: we employ a hybrid private key for each user, in which an AND gate is involved to connect and bound the identity component and the time component. Furthermore, we propose another construction which is provable secure under the recently formulized Refereed Delegation of Computation model. Finally, we provide extensive experimental results to demonstrate the efficiency of our proposed construction.

TITLE 8: Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption

AUTHOR: M. Li, S. Yu, Y. Zheng, K. Ren, W. Lou

Personal health record (PHR) is an emerging patient-centric model of health information exchange, which is often outsourced to be stored at a third party, such as cloud providers. However, there have been wide privacy concerns as personal health information could be exposed to those third party servers and to unauthorized parties. To assure the patients' control over access to their own PHRs, it is a promising method to encrypt the PHRs before outsourcing. Yet, issues such as risks of privacy exposure, scalability in key management, flexible access, and efficient user revocation, have remained the most important challenges toward achieving fine-grained, cryptographically enforced data access control. In this paper, we propose a novel

patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semitrusted servers. To achieve fine-grained and scalable data access control for PHRs, we leverage attribute-based encryption (ABE) techniques to encrypt each patient's PHR file. Different from previous works in secure data outsourcing, we focus on the multiple data owner scenario, and divide the users in the PHR system into multiple security domains that greatly reduces the key management complexity for owners and users. A high degree of patient privacy is guaranteed simultaneously by exploiting multiauthority ABE. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios. Extensive analytical and experimental results are presented which show the security, scalability, and efficiency of our proposed scheme.

TITLE 9:A cloud-based approach for interoperable electronic health records (EHRs)

AUTHOR: A. Bahga, V. Madiseti,

We present a cloud-based approach for the design of interoperable electronic health record (EHR) systems. Cloud computing environments provide several benefits to all the stakeholders in the healthcare ecosystem (patients, providers, payers, etc.). Lack of data interoperability standards and solutions has been a major obstacle in the exchange of healthcare data between different stakeholders. We propose an EHR system - cloud health information systems technology architecture (CHISTAR) that achieves semantic interoperability

-grained and scalable data access control for PHRs, we leverage attribute-based encryption (ABE) techniques to encrypt each patient's PHR file. Different from previous works in secure data outsourcing, we focus on the multiple data owner scenario, and divide the users in the PHR system into multiple security domains that greatly reduces the key management complexity for owners and users. A high degree of patient privacy is guaranteed simultaneously by exploiting multiauthority ABE. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under

emergency scenarios. Extensive analytical and experimental results are ally enforced data access control. In this paper, we propose a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semitrusted servers. To achieve fine-grained and scalable data access control for PHRs, we leverage attribute-based encryption (ABE) techniques to encrypt each patient's PHR file. Different from previous works in secure data outsourcing, we focus on the multiple data owner scenario, and divide the users in the PHR system into multiple security domains that greatly reduces the key management complexity for owners and users. A high degree of patient privacy is guaranteed simultaneously by exploiting multiauthority ABE. Our scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios. Extensive analytical and experimental results are presented which show the security, scalability, and efficiency of our proposed scheme.

TITLE 10:Designing cloudbased electronic health record system with attribute-based encryption

AUTHOR: F. Xhafa, J. Li, G. Zhao, J. Li, X. Chen, D. Wong

With the development of cloud computing, electronic health record (EHR) system has appeared in the form of patient-centric, in which patients store their personal health records (PHRs) at a remote cloud server and selectively share them with physicians for convenient medical care. Although the newly emerged form has many advantages over traditional client-server model, it inevitably introduces patients' concerns on the privacy of their PHRs due to the fact that cloud servers are very likely to be in a different trusted domain from that of the patients. In this paper, aiming at allowing for efficient storing and sharing PHRs and also eliminating patients' worries about PHR privacy, we design a secure cloud-based EHR system, which guarantees security and privacy of medical data stored in the cloud, relying on cryptographic primitive but not the full trust over cloud servers. Based on our proposed basic EHR system, we provide several extensions including adding searchability, supporting revocation functionality and enabling efficient local decryption, which fills the gap between theoretical proposal and practical application.

CHAPTER 3

SYSTEM ANALYSIS

3.1 PROJECT SCOPE

Achieves high level security to provide trustable computing and storage services. Provides data integrity, data confidentiality, authentication and authorization. Eliminates both internal and external security threats. Avoids both active and passive attacks in cloud network environment. Achieves different levels of security in cloud framework.

3.2 OBJECTIVE

To develop a secure cloud framework for accessing trusted computing and storage services in all levels of public cloud deployment model.

3.3 EXISTING SYSTEM

- Cloud computing security based on set of control-based technologies.
- Data level security for handling data in a secure manner.
- Platform level security for providing secure platform to process the data. Secure framework for proving trusted environment to the user, but it lacks in high level security and different levels of security, less focus on insider threat, active and passive attacks.

3.4 PROPOSED SYSTEM

- Your health care provider may be moving from paper records to electronic health records (EHRs) or may be using EHRs already.

- EHRs allow providers to use information more effectively to improve the quality and efficiency of your care, but EHRs will not change the privacy protections or security safeguards that apply to your health information.
- This project focuses on developing secure cloud framework for evolving and accessing trusted computing services in all levels of public cloud deployment model.
- Thus, eliminates both internal and external security threats.
- These results in achieving data confidentiality, data integrity, authentication and authorization, eliminating both active and passive attacks from cloud network environment.
- To develop a secure cloud framework for accessing trusted computing and storage services in all levels of public cloud deployment model.

3.5 HARD REQUIREMENTS

System : Pentium Dual Core.
 Hard Disk: 120 GB.
 Monitor : 15"LED
 Input Devices : Keyboard, Mouse
 Ram : 1GB.

3.6 SOFTWARE REQUIREMENTS

Operating system : Windows 7.
 Coding Language : Java
 Toolkit : Netbeans
 DATABASE : MySQL

3.7 SOFTWARE DESCRIPTION

3.7.1 JAVA

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has wearied with passage of time, while Java's has grown.

APPLICATION OF JAVA

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like Android
4. Embedded Systems
5. Robotics and games etc.

FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

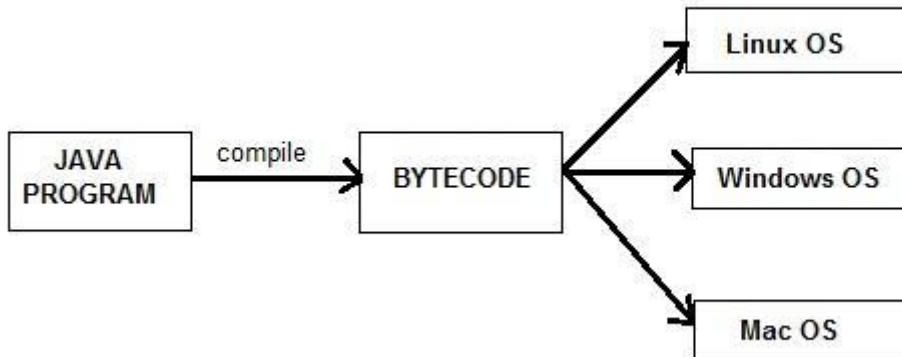
3) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime Environment can run Java Programs.



5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7) Architectural Neutral

Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

Portable

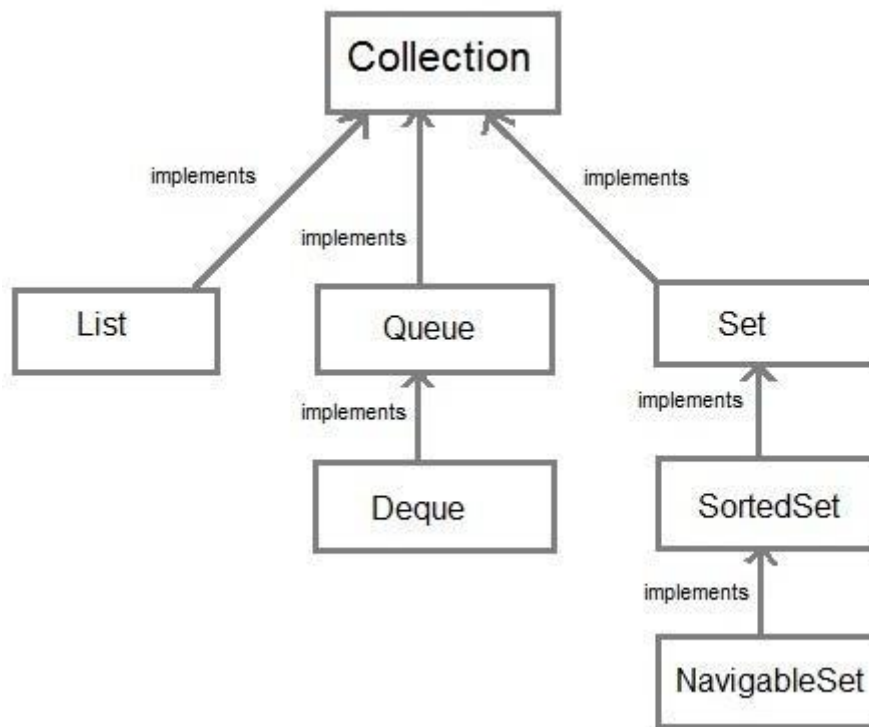
Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.

COLLECTION FRAMEWORK

Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.



3.7.2 MYSQL

MySQL, officially, but also called "My Sequel" is the world's most widely used open-source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases, though SQLite probably has more total embedded deployments. The SQL phrase stands for Structured Query Language.

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks).

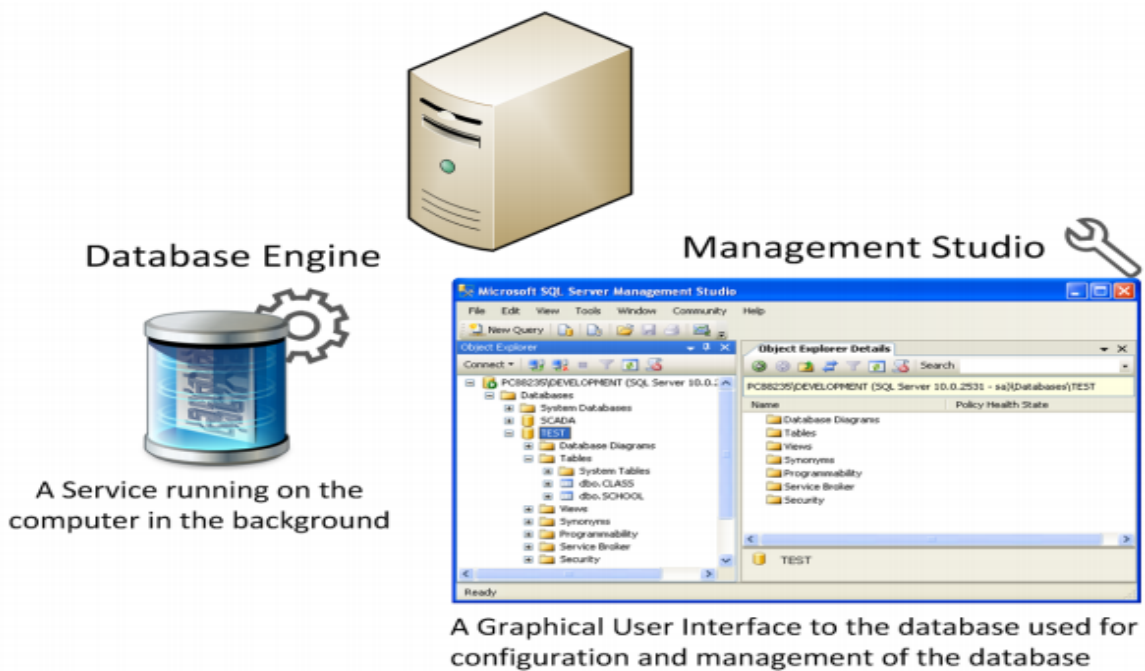
LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Freesoftware-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx,

Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr and YouTube. Microsoft is the vendor of SQL Server. We have different editions of SQL Server, where SQL Server Express is free to download and use. SQL Server uses T-SQL (Transact-SQL). T-SQL is Microsoft's proprietary extension to SQL. TSQL is very similar to standard SQL, but in addition it supports some extra functionality, built in functions, etc.

T-SQL expands on the SQL standard to include procedural programming, local variables, and various support functions for string processing, date processing, mathematics, etc. SQL Server consists of a Database Engine and a Management Studio (and lots of other stuff which we will not mention here). The Database engine has no graphical interface - it is just a service running in the background of your computer (preferable on the server). The Management Studio is graphical tool for configuring and viewing the information in the database. It can be installed on the server or on the client (or both).

SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

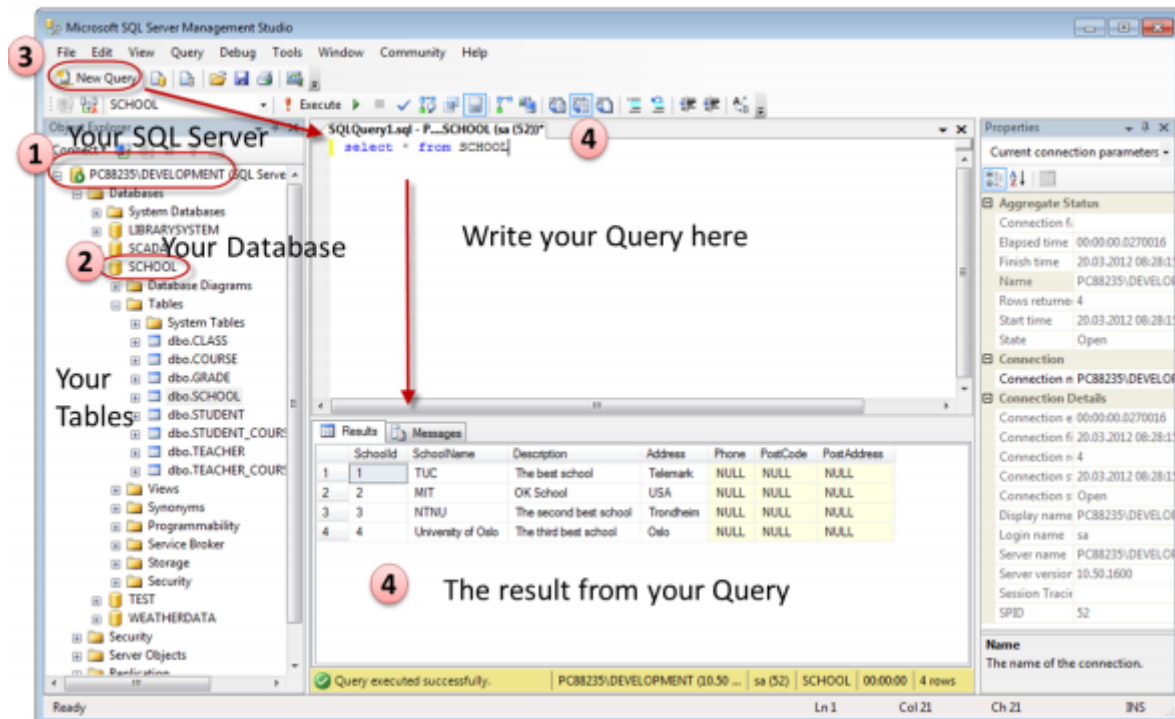


SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio is a GUI tool included with SQL Server for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools that work with objects and features of the server. As mentioned earlier, version of SQL Server Management Studio is also available for SQL Server Express Edition, for which it is known as SQL Server Management Studio Express.

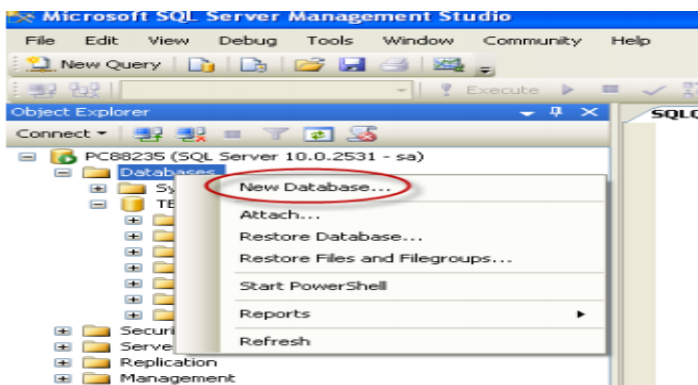
A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It can be used to visually observe and analyze query plans and optimize the database performance, among others. SQL Server Management Studio can also be used to create a new database, alter any existing database schema by adding or modifying tables and indexes, or analyze performance. It includes the query windows which provide a GUI based interface to write and execute queries.

When creating SQL commands and queries, the "Query Editor" (select "New Query" from the Toolbar) is used (shown in the figure above). With SQL and the "Query Editor" we can do almost everything with code, but sometimes it is also a good idea to use the different Designer tools in SQL to help us do the work without coding (so much).



Create a new Database

It is quite simple to create a new database in Microsoft SQL Server. Just right-click on the “Databases” node and select “New Database...”



The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects,

They are,

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

Table

A database is a collection of data about a specific topic.

Views of Table

We can work with a table in two types,

1. Design View
2. Datasheet View

Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

Query

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited). Each time we run query, we get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost

estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

The feasibility study investigates the problem and the information needs of the stakeholders. It seeks to determine the resources required to provide an information systems solution, the cost and benefits of such a solution, and the feasibility of such a solution.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity.

3.7.3SYSTEM DESIGN AND TESTING PLAN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

What data should be given as input?

How the data should be arranged or coded?

The dialog to guide the operating personnel in providing input.

Methods for preparing input validations and steps to follow when error occur.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and

also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the

- Future.

- Signal important events, opportunities, problems, or warnings.

- Trigger an action.

- Confirm an action.

Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified

requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of

the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing

Acceptance Testing

Acceptance testing is often done by the customer to ensure that the delivered product meets the requirements and works as the customer expected. It falls under the class of black box testing.

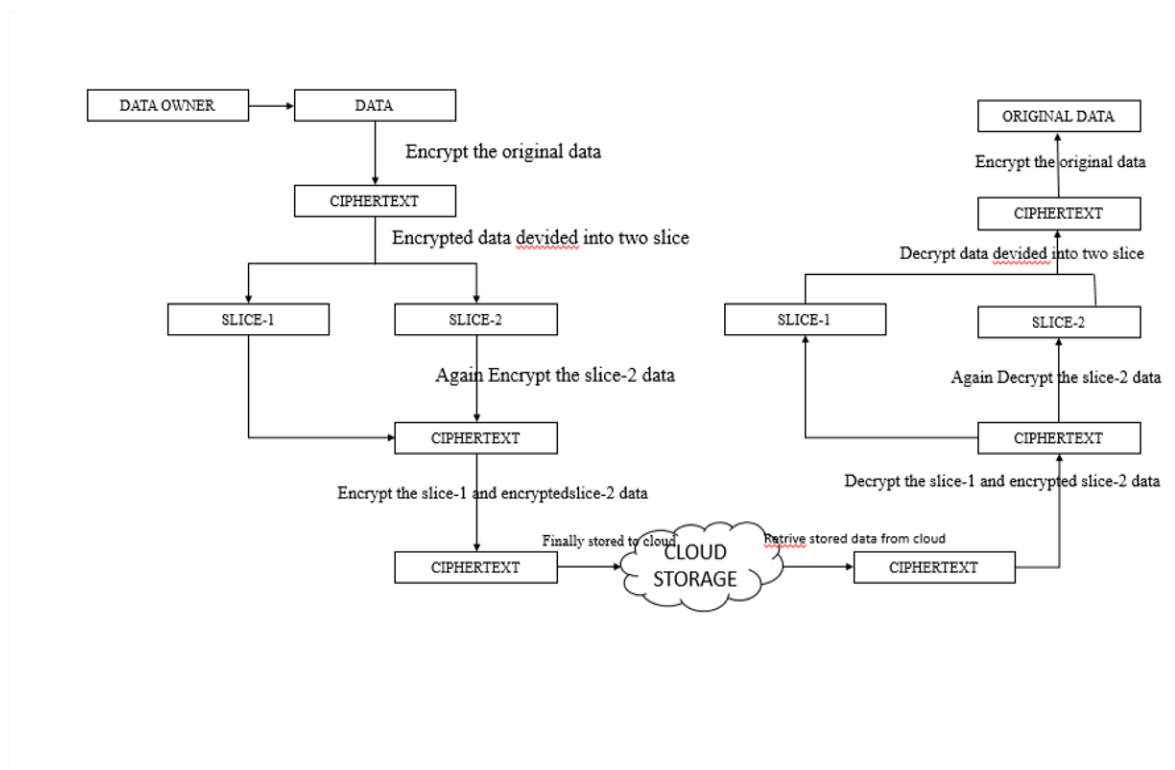
Regression Testing

Regression testing is the testing after modification of a system, component, or a group of related units to ensure that the modification is working correctly and is not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

CHAPTER 4

METHODOLOGY AND ALGORITHMS USED

4.1 SYSTEM ARCHITECTURE



4.2 PROPOSED ALGORITHMS

4.2.1 SHA ALGORITHM

Step 1: Append Padding Bits....

Message is "padded" with a 1 and as many 0's as necessary to bring the message length to 64 bits fewer than an even multiple of 512.

Step 2: Append Length....

64 bits are appended to the end of the padded message. These bits hold the binary format of 64 bits indicating the length of the original message.

Step 3: Prepare Processing Functions....

SHA1 requires 80 processing functions defined as:

$$f(t;B,C,D) = (B \text{ AND } C) \text{ OR } ((\text{NOT } B) \text{ AND } D) \quad (0 \leq t \leq 19)$$

$$f(t;B,C,D) = B \text{ XOR } C \text{ XOR } D \quad (20 \leq t \leq 39)$$

$$f(t;B,C,D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D) \quad (40 \leq t \leq 59)$$

$$f(t;B,C,D) = B \text{ XOR } C \text{ XOR } D \quad (60 \leq t \leq 79)$$

Step 4: Prepare Processing Constants....

SHA1 requires 80 processing constant words defined as:

$$K(t) = 0x5A827999 \quad (0 \leq t \leq 19)$$

$$K(t) = 0x6ED9EBA1 \quad (20 \leq t \leq 39)$$

$$K(t) = 0x8F1BBCDC \quad (40 \leq t \leq 59)$$

$$K(t) = 0xCA62C1D6 \quad (60 \leq t \leq 79)$$

Step 5: Initialize Buffers....

SHA1 requires 160 bits or 5 buffers of words (32 bits):

$$H0 = 0x67452301$$

$$H1 = 0xEFCDAB89$$

$$H2 = 0x98BADCFE$$

$$H3 = 0x10325476$$

$$H4 = 0xC3D2E1F0$$

Step 6: Processing Message in 512-bit blocks (L blocks in total message)....

This is the main task of SHA1 algorithm which loops through the padded and appended message in 512-bit blocks.

Input and predefined functions:

$M[1, 2, \dots, L]$: Blocks of the padded and appended message

$f(0;B,C,D), f(1;B,C,D), \dots, f(79;B,C,D)$: 80 Processing Functions $K(0), K(1), \dots,$

$K(79)$: 80 Processing Constant Words

$H0, H1, H2, H3, H4, H5$: 5 Word buffers with initial values

4.2.2 Advance Encryption Algorithm (AES)

(Advanced Encryption Standard), is the new encryption standard recommended by NIST to replace DES. Brute force attack is the only effective attack known against it, in which the attacker tries to test all the characters combinations to unlock the encryption. Both AES and DES are block ciphers. It has variable key length of 128, 192, or 256 bits; default 256. It encrypts data blocks of 128 bits in 10, 12 and 14 round depending on the key size. AES Encryption is fast and flexible; it can be implemented on various platforms especially in small devices. Also, AES has been carefully tested for many security applications.

Major advantages of AES over DES are

1. Data block size is 128 bits.
2. Key size 128/192/256 bits depending on version.
3. Most CPUs now include hardware AES support making it very fast.
4. It uses substitution and permutations.
5. Possible keys are 2^{128} , 2^{192} and 2^{256} [10]
6. More secure than DES.

7. Most adopted symmetric encryption algorithms.

4.2.3 Data Encryption Standard (DES)

This stands for Data Encryption Standard and it was developed in 1977. It was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). DES is 64 bits key size with 64 bits block size. Since that time, many attacks and methods have witnessed weaknesses of DES, which made it an insecure block cipher.

Algorithm:

```
function DES_Encrypt (M, K)
where M = (L, R)
M ← IP (M)
For round ← 1 to 16 do
K ← SK (K, round)
L ← Lxor F(R, Ki)
swap(L, R)
end
swap (L, R)
M ← IP-1 (M)
return M
End
```

4.2.4 chunking technique for deduplication

Chunking is a process to split a file into smaller files called chunks. In some applications, such as remote data compression, data synchronization, and data de duplication, chunking is important because it determines the duplicate detection performance of the system. Content-defined chunking (CDC) is a method to split files into variable length chunks, where the cut points are defined by some internal features of the files. Unlike fixed-length chunks, variable-length chunks are more resistant to byte shifting. Thus, it increases the probability of finding duplicate chunks within a file and between files. However, CDC algorithms require additional computation to find the cut points which might be computationally expensive for some applications. In our previous work (Widodo et al., 2016), the hash-based CDC algorithm used in the system took more process time than other processes in the de duplication system. This proposed work shows high throughput hash-less chunking . Instead of using hashes, RAM uses bytes value to declare the cut points. The algorithm utilizes a fix-sized window and a variable-sized window to find a maximum-valued byte which is the cut point. The maximum-valued byte is included in the chunk and located at the boundary of the chunk. This configuration allows RAM to do fewer comparisons while retaining the CDC property. We compared RAM with existing hash-based and hash-less deduplication systems. The experimental results show that our proposed algorithm has higher throughput and bytes saved per second compared to other chunking algorithms.

4.3 MODULES

MODULE 1: Login/ Registration

MODULE 2: Creation Storage and Instance

MODULE 3: Data Protection

MODULE 4: Data Recovery Module

4.3.1 LOGIN MODULE

This is the first activity that opens when user open the website. User needs to provide a correct contact number and a password, which user enters while registering, in order to login into the website. If information provided by the user matches with the data in the database table then user successfully login into the website else message of login failed is displayed and user need to reenter correct information. A link to the register activity is also provided for registration of new users.

4.3.2 REGISTRATION MODULE

A new user who wants to access the webpage needs to register first before login. By clicking on register button in login activity, the register activity gets open. A new user registers by entering full name, password and contact number. A user needs to enter password again in confirm password textbox for confirmation. When user enters the information in all textboxes, on the click of register button, the data is transferred to database and user is directed to login activity again. Registered user then needs to login in order to access the webpage. Validations are applied on all the textboxes for proper functioning of the webpage. Like information in each textbox is must that is each textbox, either it is of name, contact, password or confirm password, will not be empty while registering. If any such textbox is empty app will give message of information is must in each textbox. Also data in password and confirm

password fields must match for successful registration. Another validation is contact number must be valid one that is of 10 digits. If any such validation is violated then registration will be unsuccessful and then user needs to register again. Message that webpage will display when one of the field is empty. If all such information is correct user will be directed to login activity for login into the webpage.

4.3.3 CREATION STORAGE AND INSTANCE

The data owner has not control over the data after it is uploaded on cloud. In this module, the original data get encrypted into two different values. The data in each slice can be encrypted by using different cryptographic algorithm's and encryption key before storing them in the Cloud.

4.3.4 DATA PROTECTION

In this module, the technique is to store data in a proper secure and safe manner in order to avoid intrusions and data attacks meanwhile it will reduce the cost and time to store the encrypted data in the Cloud Storage.

4.3.5 DATA RECOVERY MODULE

In this Module, User can recover the data from cloud server using different types of techniques.

CHAPTER 5

RESULTS AND DISCUSSION

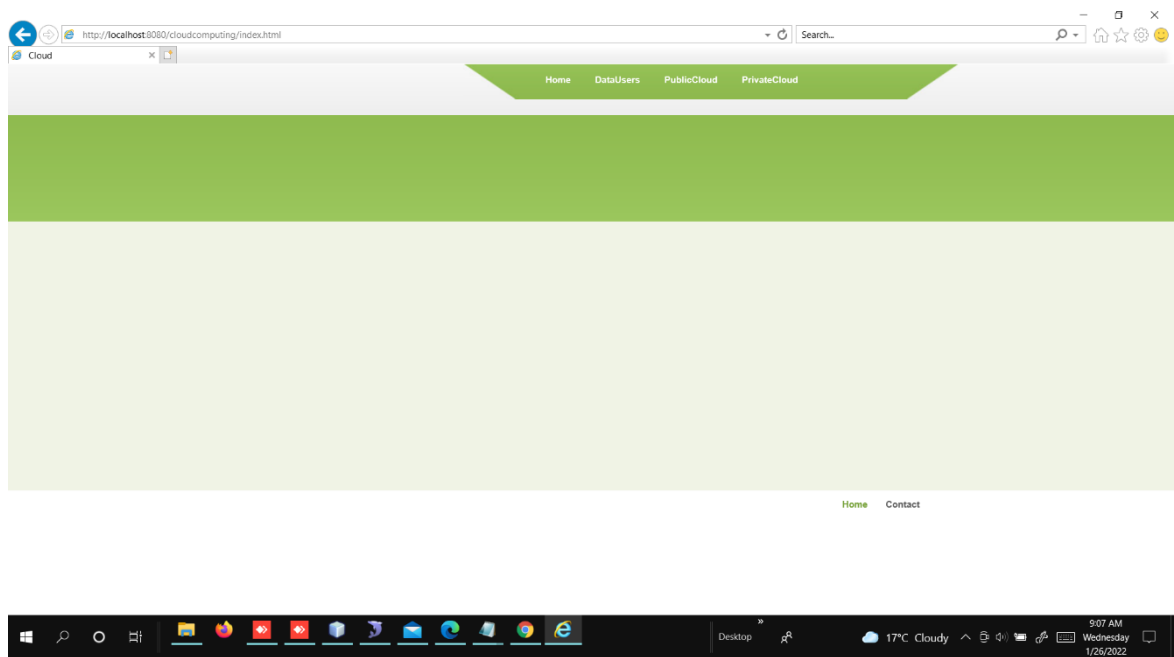


Figure 5.1 Home page

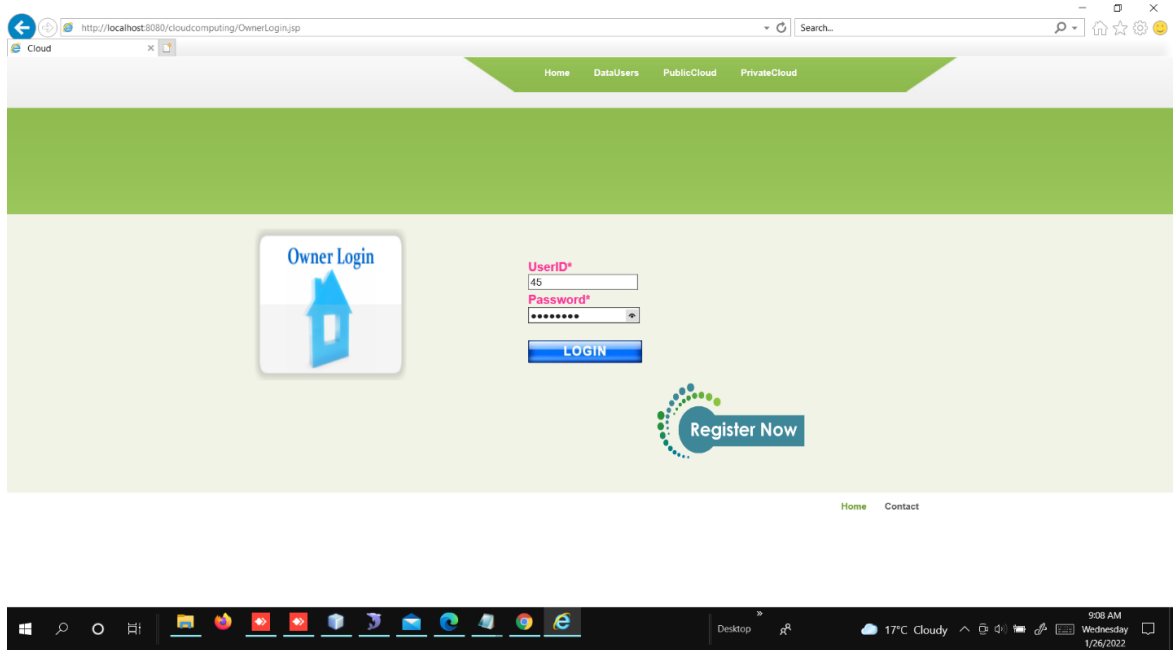


Figure 5.2 Login page

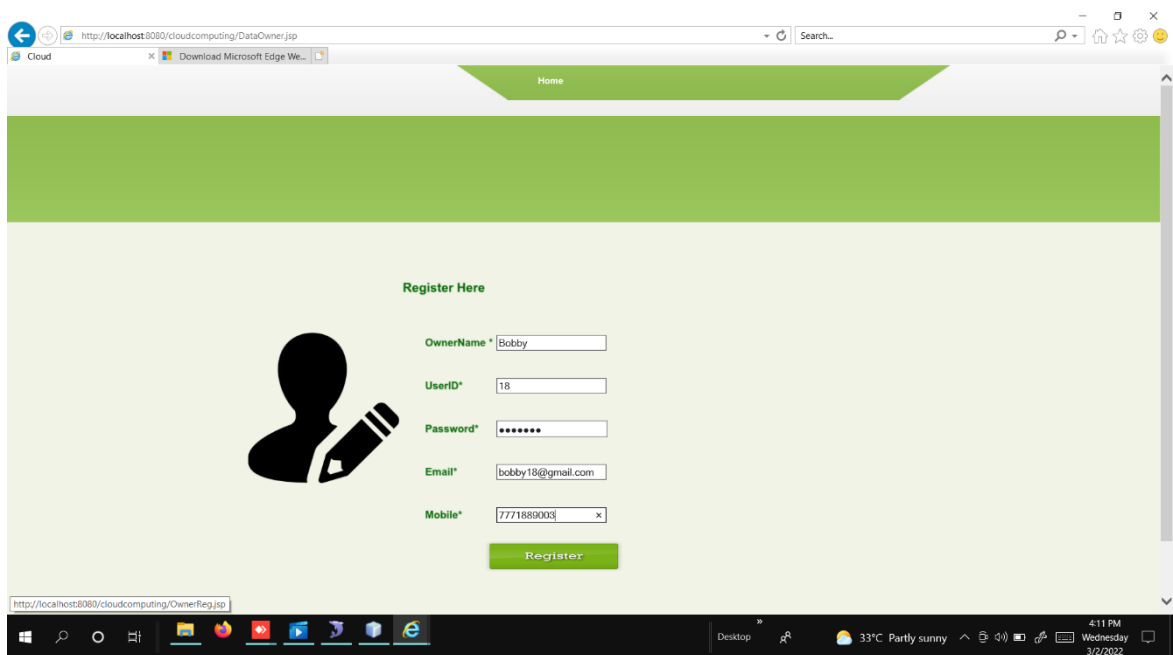


Figure 5.3 Registration page

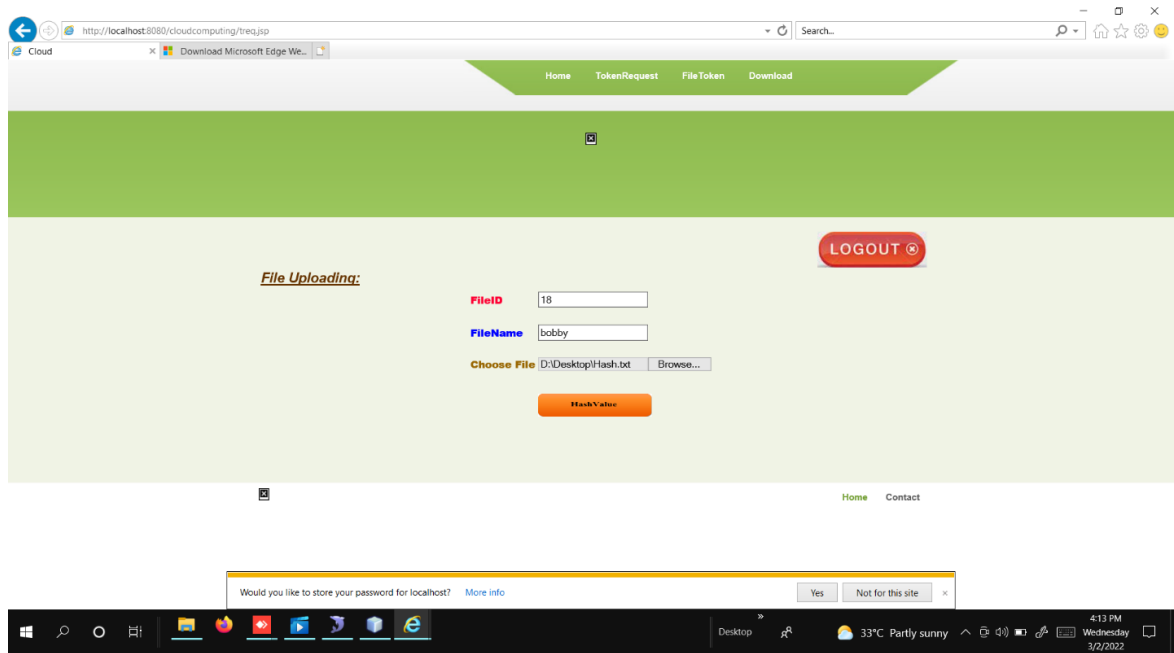


Figure 5.4 File upload page

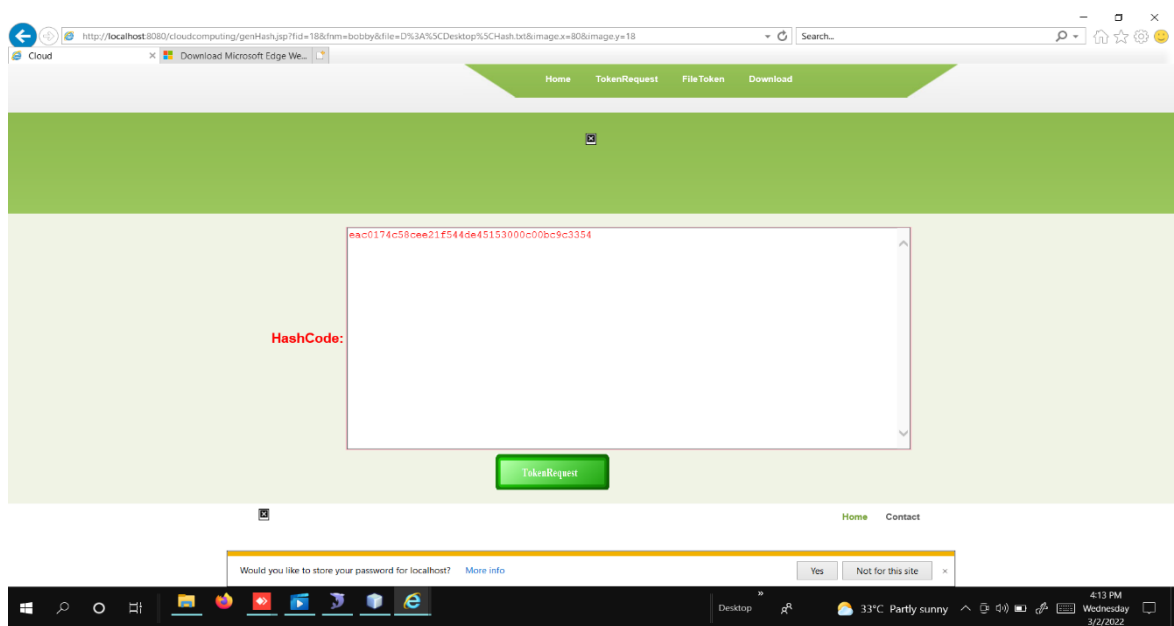


Figure 5.5 Token request and hash generation

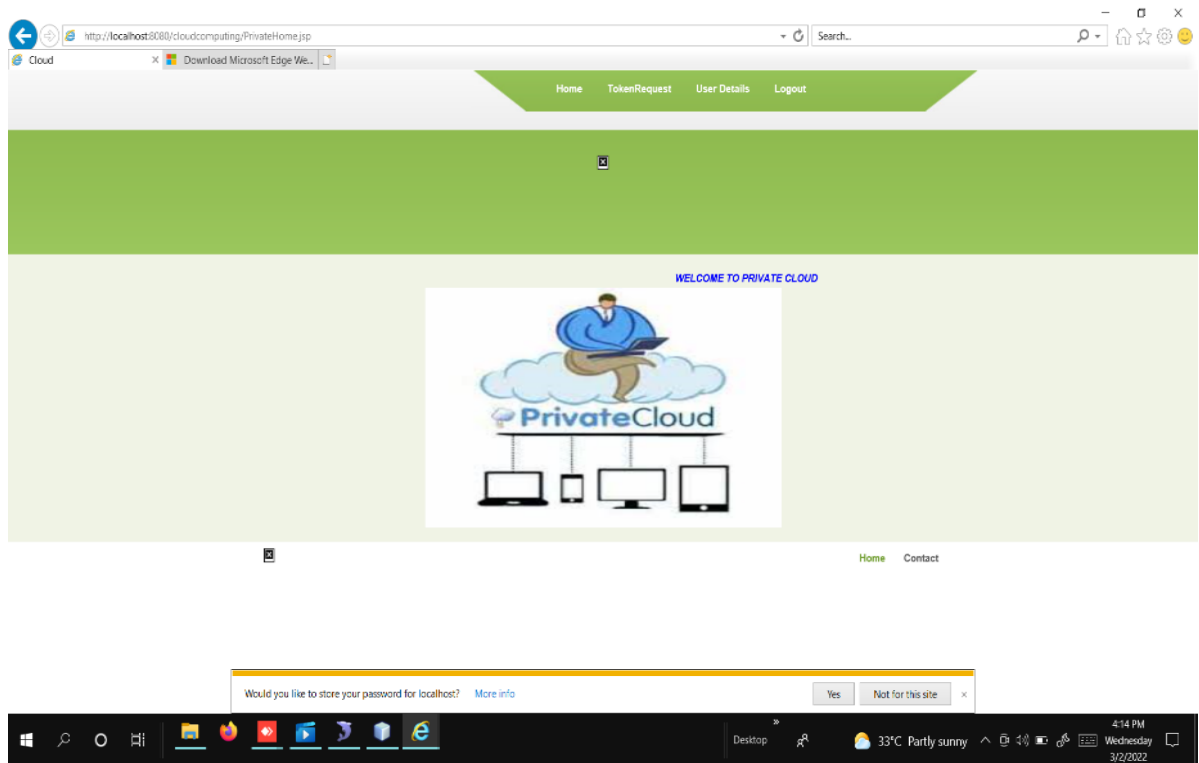


Figure 5.6 Private cloud home page

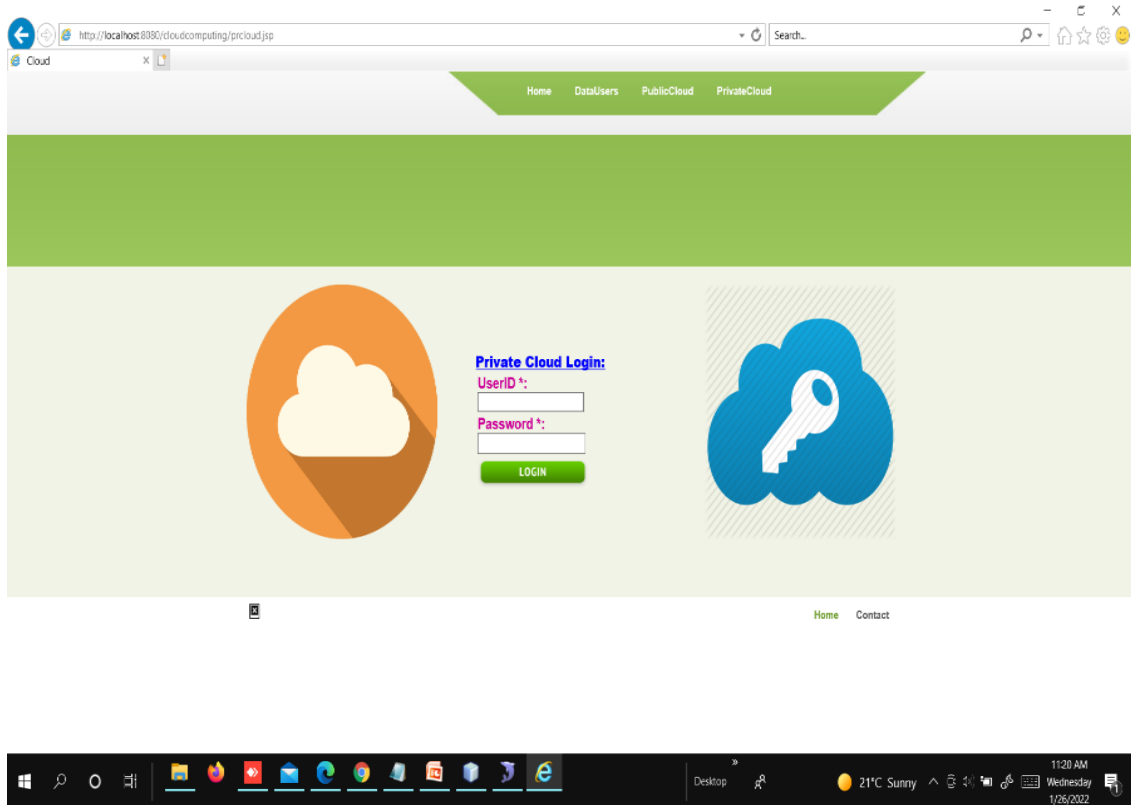


Figure 5.7

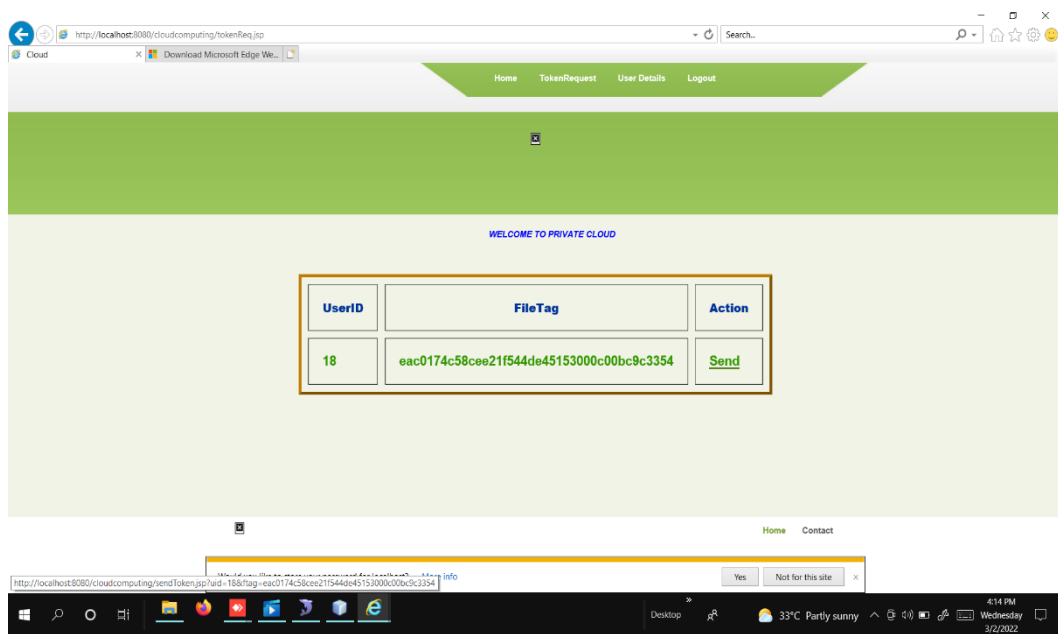


Figure 5.8

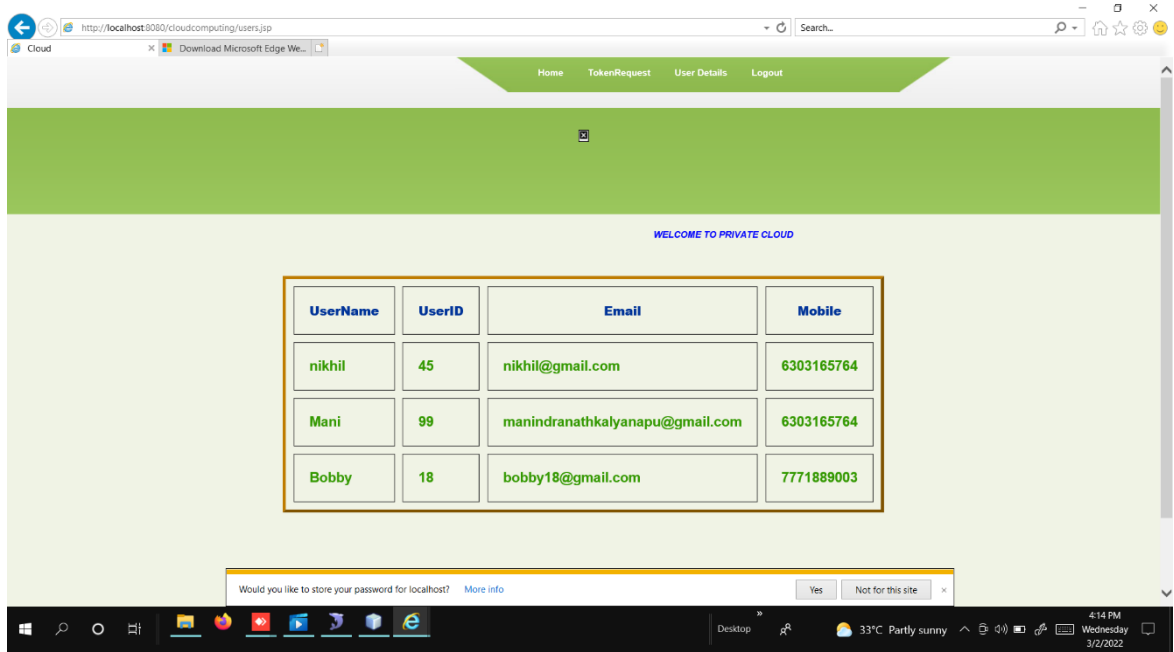


Figure 5.9

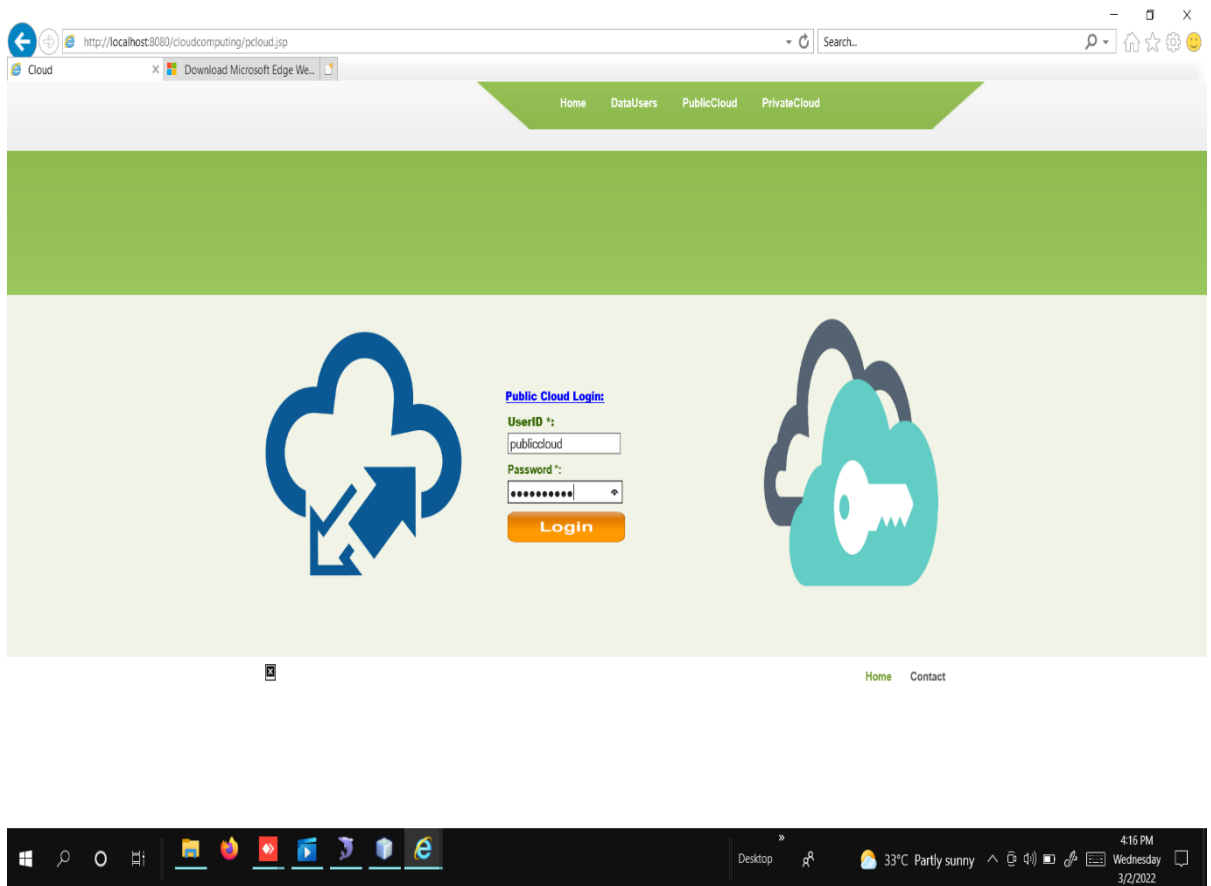


Figure 5.10

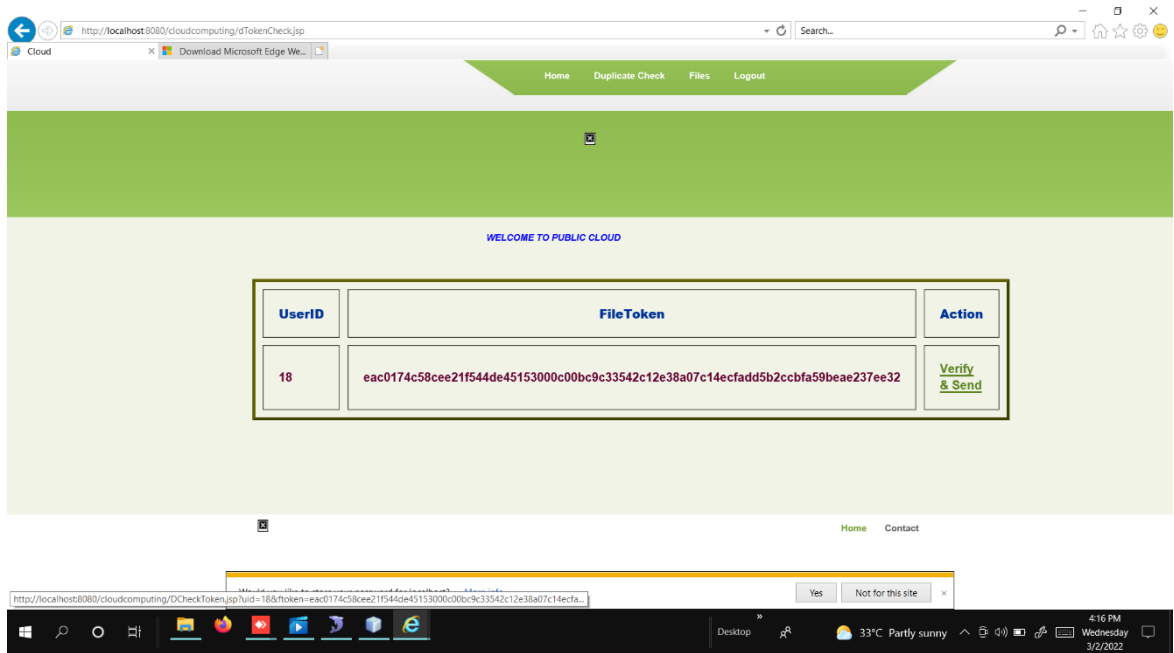


Figure 5.11

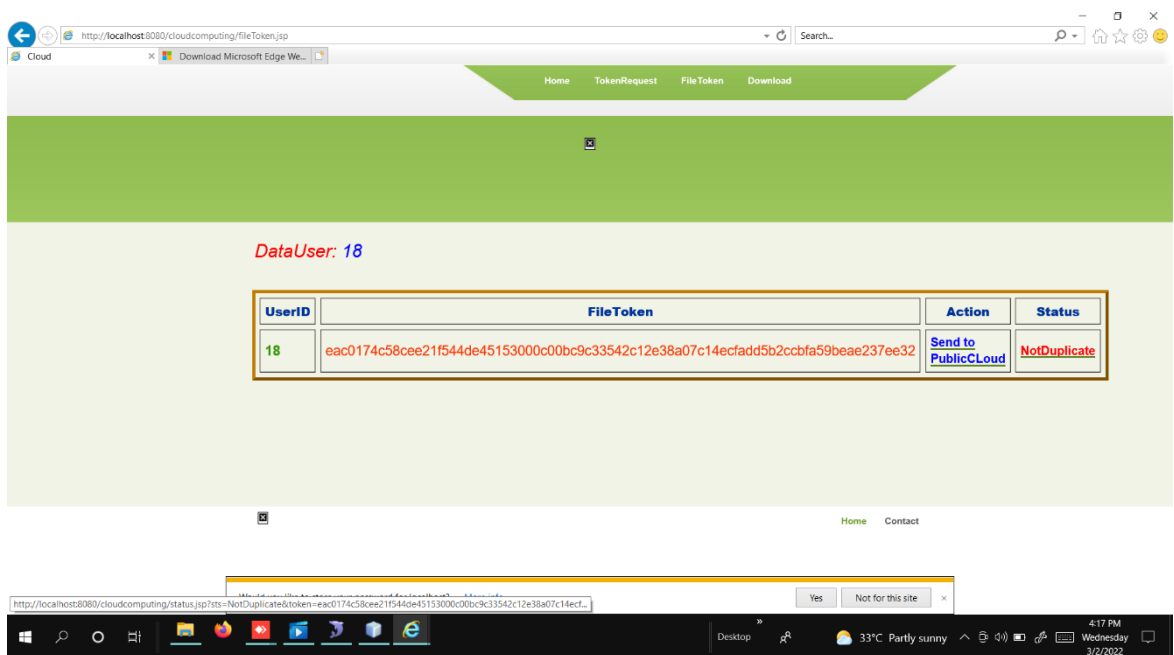


Figure 5.12

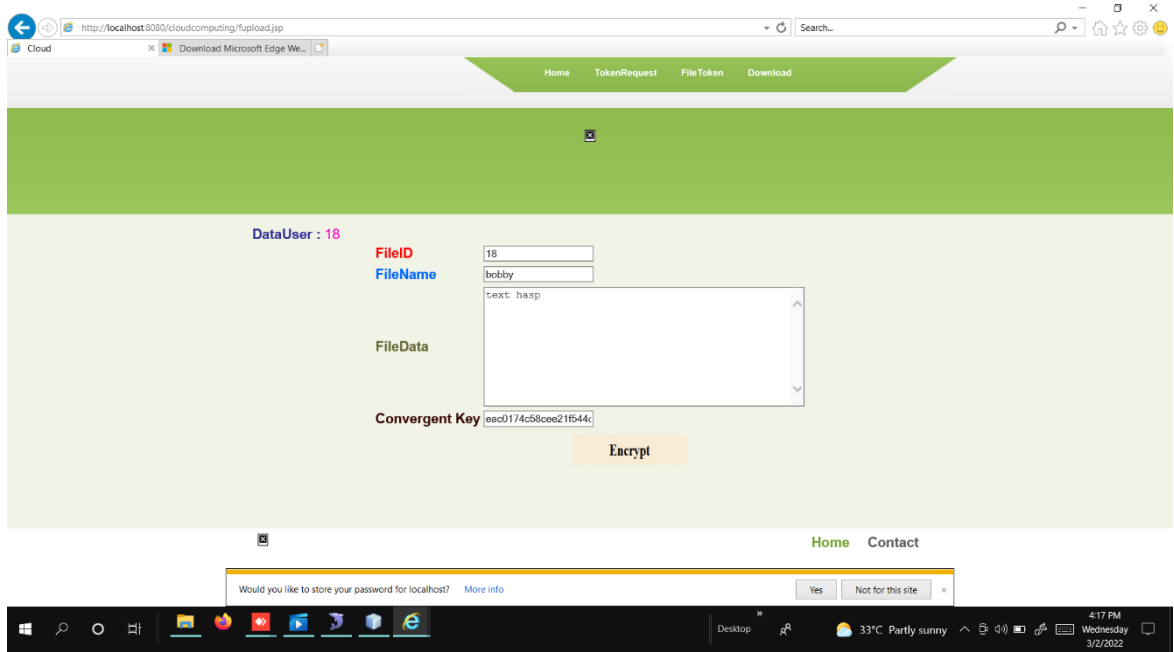


Figure 5.13

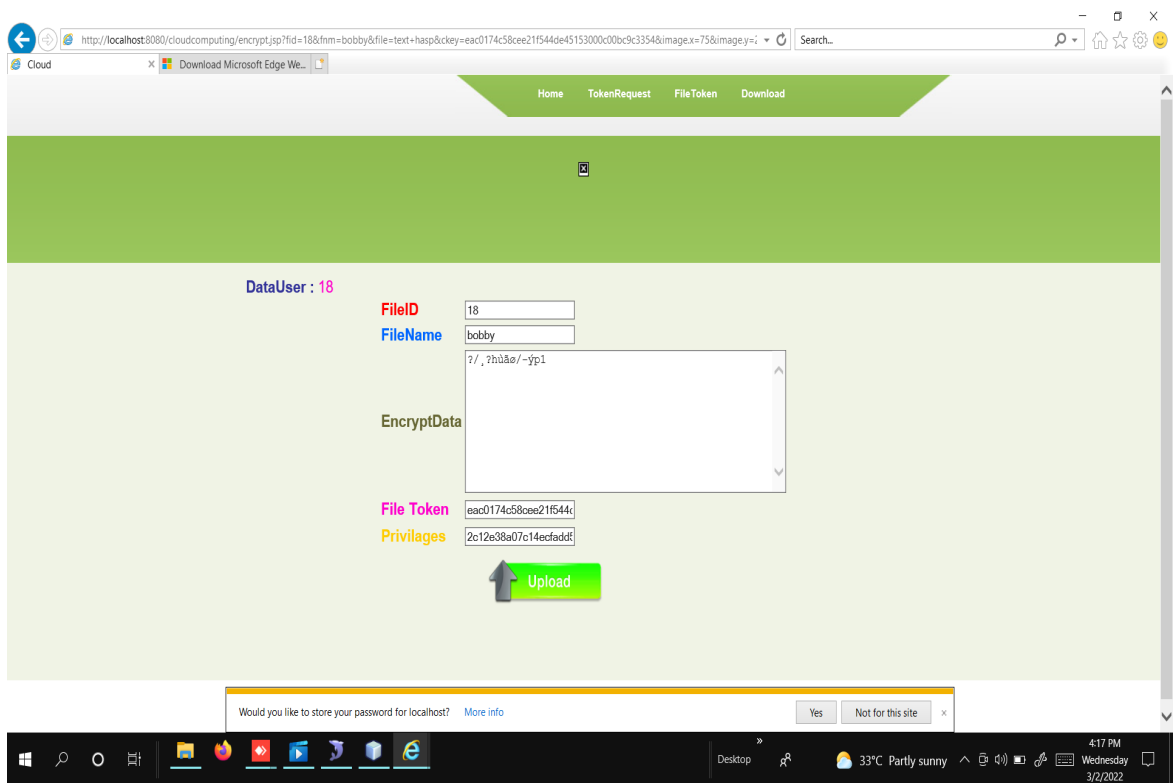


Figure 5.14

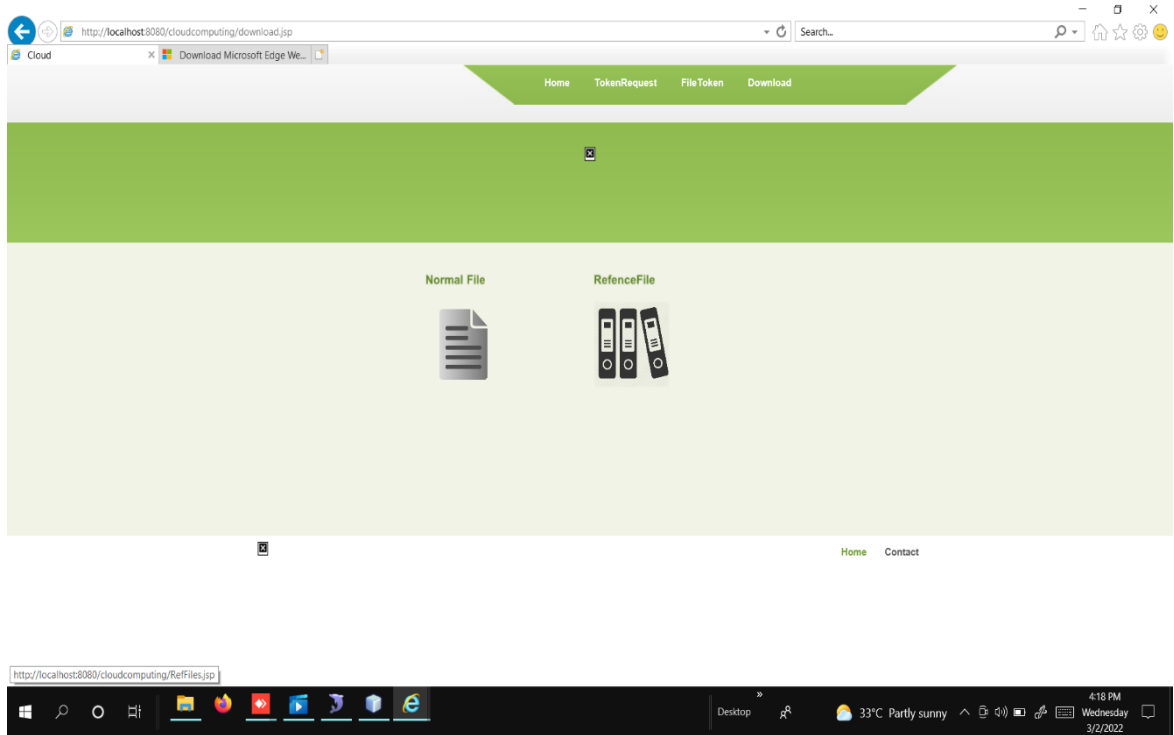


Figure 5.15

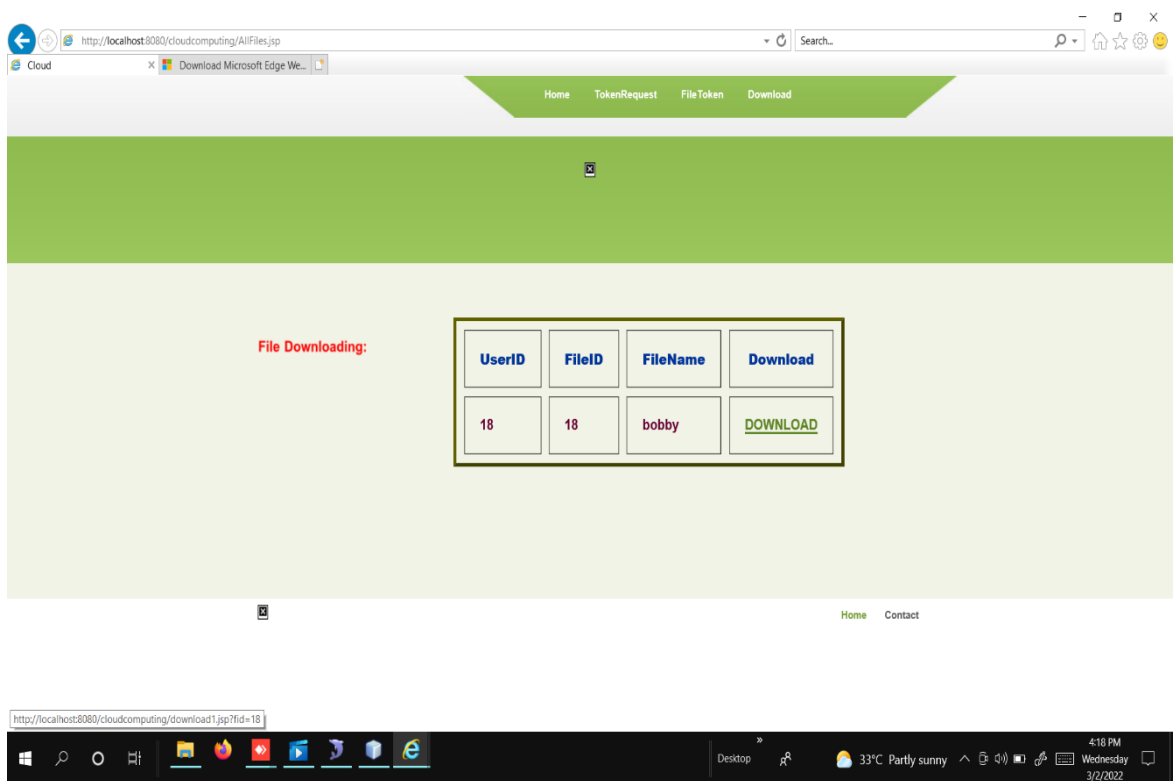


Figure 5.16

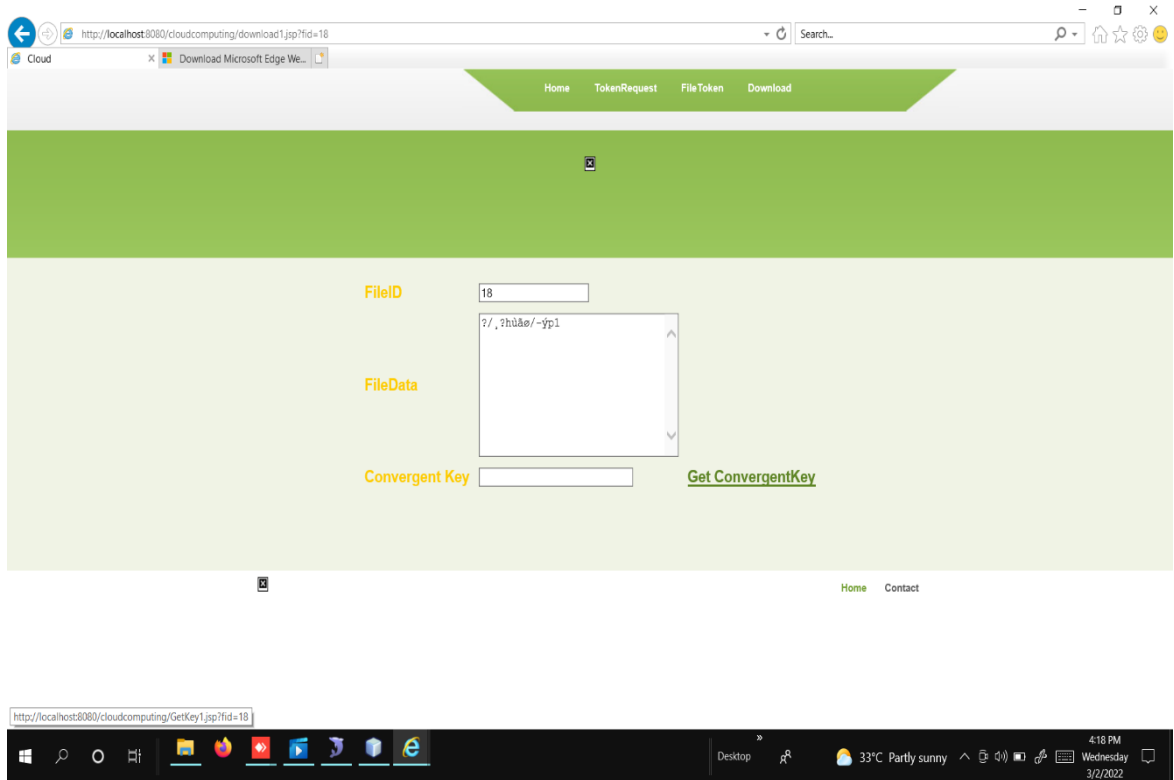


Figure 5.17

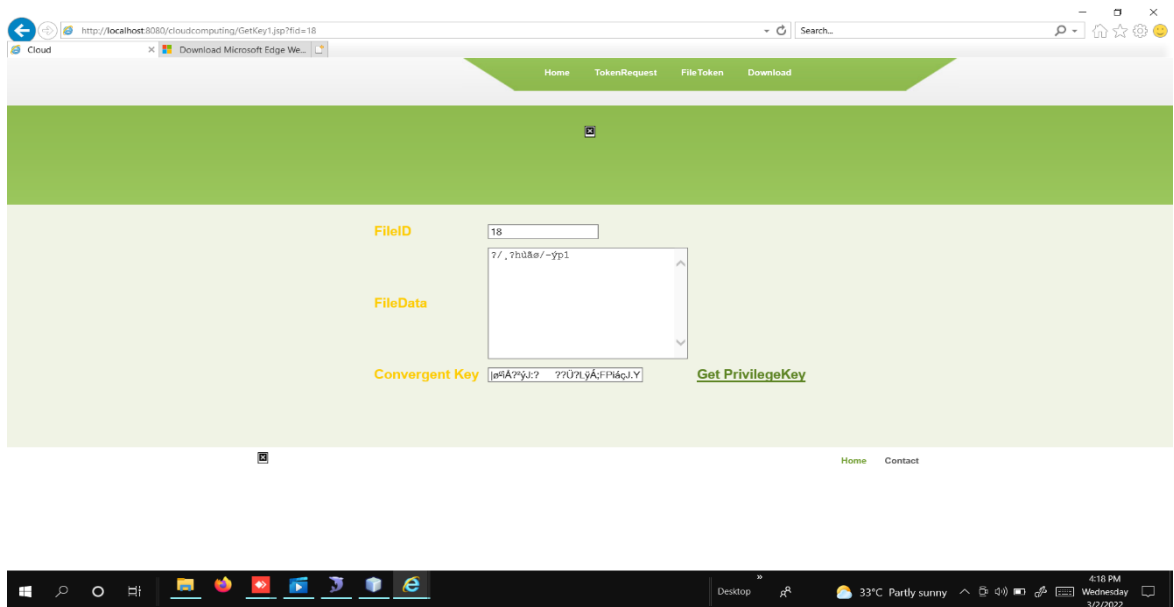


Figure 5.18

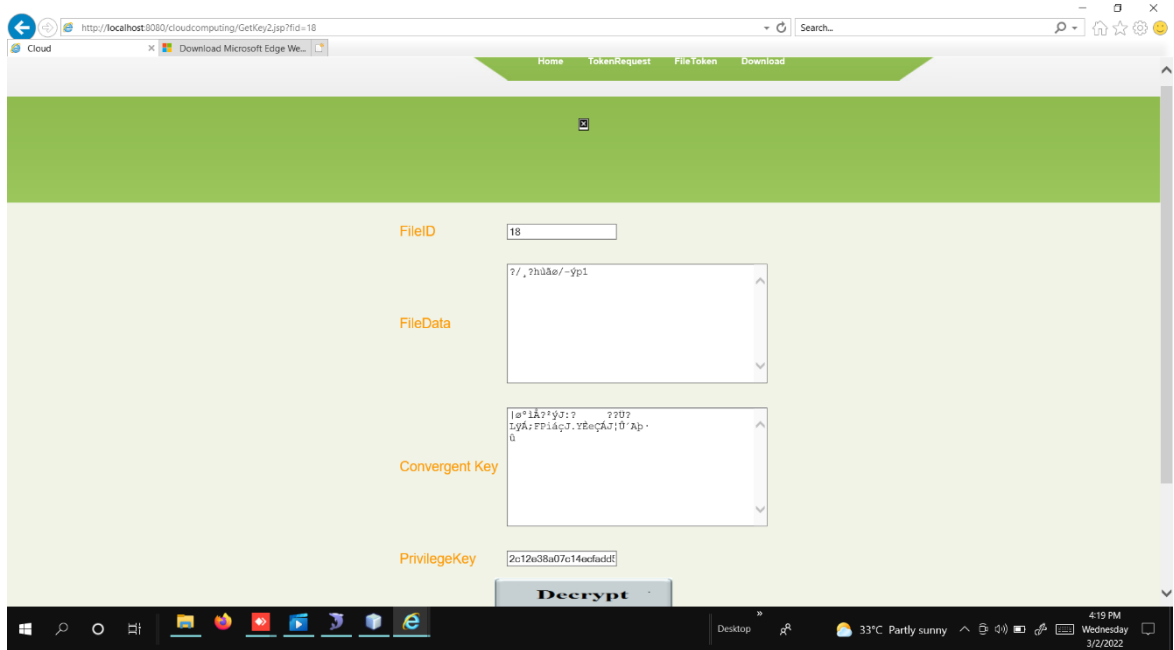


Figure 5.19

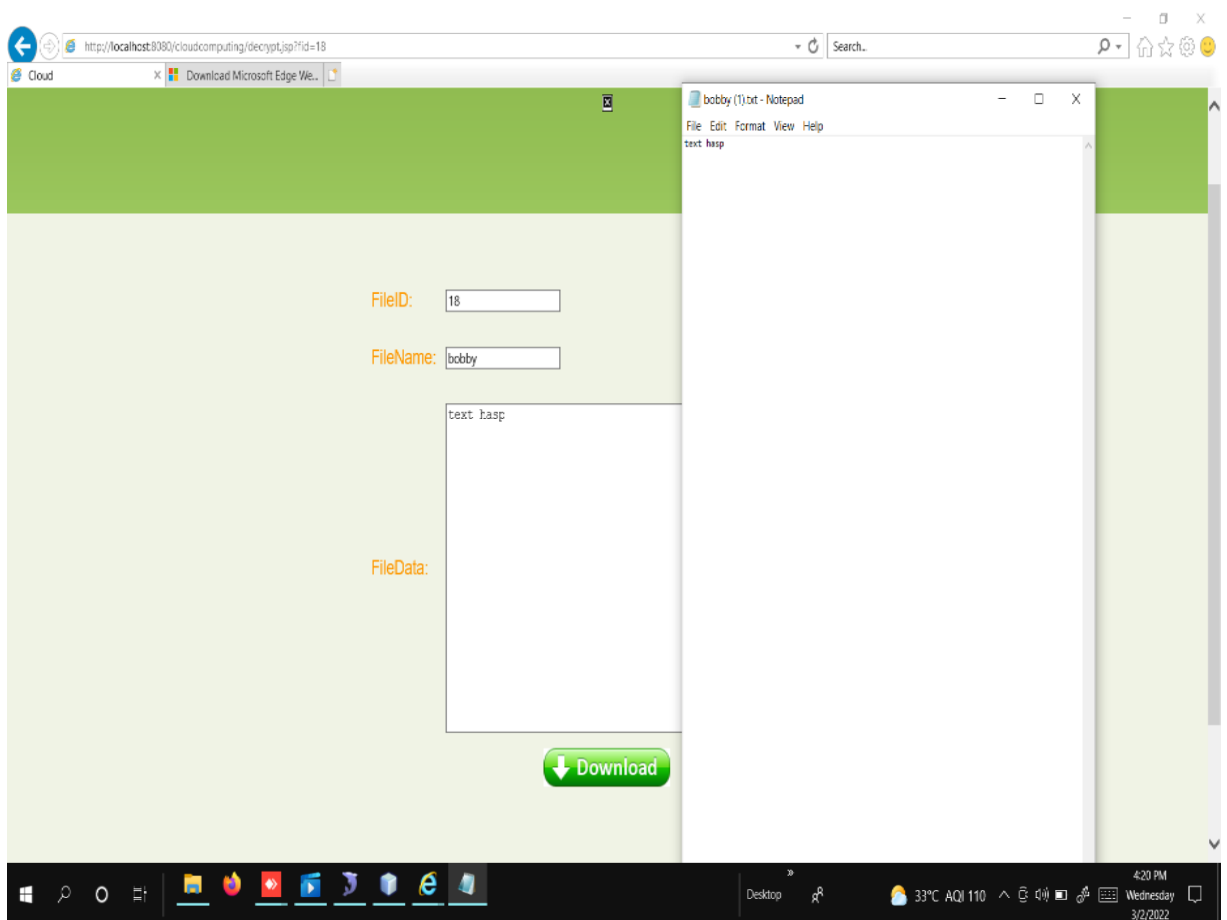


Figure 5.20

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

- These results in achieving data confidentiality, data integrity, authentication and authorization, eliminating both active and passive attacks from cloud network environment.
- To develop a secure cloud framework for accessing trusted computing and storage services in all levels of public cloud deployment model.

REFERENCES

- [1] C. Chu, S. Chow, W. Tzeng, J. Zhou, R. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp.468-477, Feb. 2014.
- [2] Y. Tong, J. Sun, S. Chow, P. Li, "Cloud-assisted mobile-access of health data with privacy and auditability", IEEE Journal of Biomedical and Health Informatics, vol. 18, no. 2, Mar. 2014.
- [3] Z. Pervez, A. Khattak, S. Lee, Y. Lee, "SAPDS: Self-healing attributebased privacy aware data sharing in cloud", The Journal of Supercomputing, vol. 62, no. 1, pp. 431-460, Oct. 2012.
- [4] C. Fan, V. Huang, H. Rung, "Arbitrary-state attribute-based encryption with dynamic membership", IEEE Transactions on Computers, vol. 63, no. 8, pp. 1951-1961, Apr. 2013.
- [5] D. Boneh, G. Crescenzo, R. Ostrovsky, G. Persiano, "Public key encryption with keyword search", in Eurocrypt 2004, Interlaken, Switzerland, May 2-6, 2004, pp.506-522.
- [6] N. Cao, C. Wang, M. Li, K. Ren, W. Lou, "Privacy-preserving multikeyword ranked search over encrypted cloud data", IEEE Transactions on Parallel and

Distributed Systems, vol. 25, no. 1, pp. 222-233, Nov. 2013.

[7] S. Seo, M. Nabeel, X. Ding, E. Bertino, "An efficient certificateless encryption for secure data sharing in public clouds", IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 9, pp. 2107-2119, Sept. 2014.

[8] L.A. Dunning, R. Kresman, "Privacy preserving data sharing with anonymous ID assignment", IEEE Transactions on Information Forensics and Security, vol. 8, no. 2, pp.402-413, Feb. 2013.

[9] X. Chen, X. Huang, J. Li, J. Ma, D. Wong, W. Lou, "New algorithms for secure outsourcing of large-scale systems of linear equations", IEEE Transactions on Information and Forensics Security, vol. 10, no. 1, pp. 69- 78, Jan. 2015.

[10] X. Chen, J. Li, J. Weng, J. Ma, W. Lou, "Verifiable computation over large database with incremental updates" IEEE Transactions on Computers, vol. 65, no. 10: 3184-3195, Oct. 2016. [11] C. Gao, Q. Cheng, X. Li, S. Xi

APPENDIX

A. SOURCE CODE

AES:

```
3  <%! String data1=null;%>
4  <%
5  String u =(String)session.getAttribute("uid");
6  String fid=request.getParameter("fid");
7  String ckey=request.getParameter("ckey");
8  System.out.println("ckey="+ckey);
9  session.setAttribute("ckey",ckey);
10 Connection con=databasecon.getconnection();
11 Statement st=con.createStatement();
12 Statement st2=con.createStatement();Statement st3=con.createStatement();
13
14 ResultSeti=st.executeQuery("select *from filedata");
15 if(i.next())
16 { String token=null,pkey=null;
```

```

17  byte b[]=i.getBytes(3);
18  //String data=new String(b);
19  //byte b2[]=data.getBytes();
20  PreparedStatementps=con.prepareStatement("update      filedata      set
      file_=AES_ENCRYPT(?,?,"+ckey+"") where fid='"+fid+"'");
21  ps.setBytes(1,b);
22  ps.executeUpdate();
23  ResultSet i1=st3.executeQuery("select *from filedata");
24  if(i1.next())
25  {
26  byte b1[]=i1.getBytes(3);
27  data1=new String(b1);
28  }
29  Statement st1=con.createStatement();
30  ResultSet r=st1.executeQuery("select token from token where uid='"+uid+"' ");
31  if(r.next())
32  {
33  token=r.getString(1);
34  }
35  ResultSet  r2=st2.executeQuery("select  pkey  from  signup  where
      userid='"+uid+"' ");
36  if(r2.next())
37  {
38  pkey=r2.getString(1);
39
40  }
41
42  %>
43
44  <center>

```

```

45 <form action="fupload1.jsp">
46     <table align=center>
47
48     <tr>
49         <td><font color="red" size=+1><b>FileID</td><td><input type="text"
        name="fid" value=<%=i.getString(1)%>></td>
50     </tr><tr></tr>
51     <tr>
52         <td><font color="0066ff" size=+1><b>FileName</td><td><input
        type="text" name="fnm" value=<%=i.getString(2)%>></td>
53     </tr><tr></tr>
54
55 <tr>
56     <td><font color="666633" size=+1><b>EncryptData</td>
57     <td><textarea name="file" cols="50"
        rows="10"><%=data1%></textarea></td>
58 </tr><tr></tr>
59
60 <tr>
61 <tr><td><font color="ff00cc" size=+1><b>File Token</td><td><input
        type="text" name="token" value=<%=token%>></td></tr><tr></tr><tr></tr>
62 <tr><td><font color="ffcc00" size=+1><b>Privileges</td><td><input
        type="text" name="pkey"
        value=<%=pkey%>></td></tr><tr></tr><tr></tr><tr></tr><tr></tr>
63
64 </table>
65
66 <tr>
67
68 <table align="center">

```

```

69 <tr><td      width="60%"><input      type="image"      src="images/up.png"
      name="image" width="150" height="50"><center></center></td></tr>
70
71
72 </table>
73
74
75
76 </table>
77 </form><%}%><br><br><br>
      <%@ include file="Ownerfooter.jsp"%>

```

DES:

```

78 <%
79 String pkey=null,ckey=null;
80 Connection con=databasecon.getConnection();
81 Statement st=con.createStatement();
82 Statement st1=con.createStatement();
83 Statement st2=con.createStatement();
84 String fid=request.getParameter("fid");
85 System.out.println("fid="+fid);
86 ResultSettr=st1.executeQuery("select  pkey  from  servicecloud  where
      fid='"+fid+"'");
87 if(rr.next())
88 {
89 pkey=rr.getString(1);
90 System.out.println("pkey="+pkey);
91
92 }
93 ResultSet  rr1=st2.executeQuery("select  AES_DECRYPT(ckey,'" +pkey+"")

```

```

        from servicecloud where fid='"+fid+"'");
94  if(rr1.next())
95  {
96  ckey=rr1.getString(1);
97  System.out.println("ckey="+ckey);
98  }
99  ResultSeti=st.executeQuery("select
        fid,fname,AES_DECRYPT(fdata,'" +ckey+"')from        servicecloud        where
        fid='"+fid+"'");
100 System.out.println("i="+i);
101 if(i.next())
102 { System.out.println("i=");
103     byte b[]=i.getBytes(3);
104     System.out.println("i="+b);
105     String d1=b.toString();
106     System.out.println("d1="+d1);
107     String data=new String(b);
108     %><br><br>
109     <font size="4" color="ff9900">
110     <table cellpadding=10 align="center">
111 <td>FileID:</td><td><input                type="text"                name="fid"
        value=<%=fid%>></td></tr><tr></tr><tr></tr>
112 <tr><td>FileName:</td><td><input                type="text"                name="fname"
        value=<%=i.getString(2)%>></td></tr><tr></tr><tr></tr>
113     <tr><td>FileData:</td>
114     <td><textarea                name="file"                cols="60%"
        rows="20%"><%=data%></textarea></td>
115 </tr></table>
116 <table align="center">
117 <tr><td><a

```

```
href="finalDownload.jsp?fid=<%=fid%>&ckey=<%=ckey%>"><imgsrc="image  
s/dwn.png" width="160" height="36" border="0" alt=""></a></td></tr>
```

118

119 </table>

120 </form><%}%>

<%@ include file="Ownerfooter.jsp"%>

SHA:

121 package Hash;

122 import java.io.UnsupportedEncodingException;

123 import java.security.MessageDigest;

124 import java.security.NoSuchAlgorithmException;

125 import java.io.*;

126 public class HMAC_SHA

127 {

128 private static String HMAC(String message, String algorithm)

129 {byte[] hashedBytes=null;

130

131 try {

132 MessageDigest digest = MessageDigest.getInstance(algorithm);

133 hashedBytes = digest.digest(message.getBytes("UTF-8"));

134

135

136 } catch (Exception ex) {

137 System.out.println(ex);

138 }

139 return convertByteArrayToHexString(hashedBytes);

140 }

141 static String convertByteArrayToHexString(byte[] arrayBytes) {

142 StringBuffer stringBuffer = new StringBuffer();

```
143     for (int i = 0; i<arrayBytes.length; i++) {  
144 stringBuffer.append(Integer.toString((arrayBytes[i] & 0xff) + 0x100, 16)  
145 .substring(1));  
146     }  
147     return stringBuffer.toString();  
148 }  
}
```

ANONYMOUS DATA SHARING SCHEME ...

ORIGINALITY REPORT

10%

SIMILARITY INDEX

8%

INTERNET SOURCES

9%

PUBLICATIONS

6%

STUDENT PAPERS

PRIMARY SOURCES

1

researchr.org

Internet Source

1%

2

Submitted to UC, Irvine

Student Paper

1%

3

Ch. Chethan Sai, Ch. Sai Prakash, Jithina Jose, Suja Cherukullapurath Mana, B Keerthi Samhitha. "Analysing Android App Privacy Using Classification Algorithm", 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020

Publication

1%

4

Submitted to Bakersfield College

Student Paper

1%

5

Huaqun Wang. "Anonymous Data Sharing Scheme in Public Cloud and Its Application in E-Health Record", IEEE Access, 2018

Publication

1%

6

api.crossref.org

Internet Source

1%

