VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgaum – 590 014

A PROJECT REPORT ON

"Privacy Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing"

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
INFORMATION SCIENCE AND ENGINEERING
BY

S Prajwal (1CR14IS082)

Syed Yunus Ahmed (1CR14IS107)

Yash Tarachand Wanvari (1CR14IS118)

Under the guidance of

Ms. Aiswarya Lakshmi
(Assistant Professor, Department of ISE, CMRIT)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING
CMR INSTITUTE OF TECHNOLOGY

BANGALORE-37

CMR INSTITUTE OF TECHNOLOGY
BANGALORE-37

Department of Information Science & Engineering

Certificate

This is to certify that the project entitled **“Privacy Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing”** has been successfully completed by **S Prajwal, 1CR14IS082, Syed Yunus Ahmed, 1CR14IS107, Yash Tarachand Wanvari, 1CR14IS118,** bonafide students of CMR Institute of Technology in partial fulfillment of the requirements for the award of degree of Bachelor of Engineering in **Information Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the academic year 2017-2018. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the project report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Ms. Aiswarya Lakshmi K Assistant proffesor Department of ISE, CMRIT	Dr. Prem Kumar Ramesh Professor & HOD Department of ISE, CMRIT	Dr.Sanjay Jain Principal, CMRIT

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

CMR INSTITUTE OF TECHNOLOGY
BANGALORE-37

Department of Information Science & Engineering

DECLARATION

We, **S Prajwal, 1CR14IS082, Syed Yunus Ahmed, 1CR14IS107, Yash Tarachand Wanvari, 1CR14IS118,** bonafide students of CMR Institute of Technology, Bangalore, hereby declare that the dissertation entitled, **“Privacy Preserving Data Encryption Strategy for Big Data in Mobile Cloud Computing”** has been carried out by us under the guidance of **Ms. Aiswarya Lakshmi K, Assistant Professor,** CMRIT, Bangalore, in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in **Information Science Engineering**, of the Visvesvaraya Technological University, Belgaum during the academic year 2017-2018. The work done in this dissertation report is original and it has not been submitted for any other degree in any university.

S Prajwal	1CR14IS082
Syed Yunus Ahmed	1CR14IS107
Yash Tarachand Wanvari	1CR14IS118

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without mentioning people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, with gratitude I acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We would like to thank **Dr. Sanjay Jain,** Principal, CMRIT, Bangalore for providing excellent academic environment in the college and his never-ending support for the B.E program.

We would like to express our gratitude towards **Dr. H.N.Shankar,** Professor and Dean Academics and Research, CMRIT, Bangalore who provided guidance and gave valuable suggestions regarding the project.

We would also like to thank **Dr. Prem Kumar Ramesh,** Professor & HOD and **Mr. Manoj Challa**, Associate Professor & Program Coordinator, Department of Information Science, CMRIT, Bangalore who shared their opinions and experiences through which we received the required information crucial for the project.

We consider it a privilege and honour to express our sincere gratitude to our internal guide **Ms. Aiswarya Lakshmi K ,** Assistant Professor, Department of Information Science & Engineering, for her valuable guidance throughout the tenure of this project work.

We would also like to thank all the faculty members who have always been very co-operative and generous. Conclusively, we also thank all the non-teaching staff and all others who have done immense help directly or indirectly during my project.

S Prajwal (1CR14IS082)
Syed Yunus Ahmed (1CR14IS107)
Yash Tarachand Wanvari (1CR14IS118)

ABSTRACT

Privacy has become a considerable issue when the applications of big data are dramatically growing in cloud computing. The benefits of the implementation for these emerging technologies have improved or changed service models and improve application performances in various perspectives. However, the remarkably growing volume of data sizes has also resulted in many challenges in practice. The execution time of the data encryption is one of the serious issues during the data processing and transmissions. Many current applications abandon data encryptions in order to reach an adoptive performance level companioning with privacy concerns. In this paper, we concentrate on privacy and propose a novel data encryption approach, which is called Dynamic Data Encryption Strategy (D2ES). Our proposed approach aims to selectively encrypt data and use privacy classification methods under timing constraints. This approach is designed to maximize the privacy protection scope by using a selective encryption strategy within the required execution time requirements.

TABLE OF CONTENT

LIST OF FIGURES

Chapter 1

INTRODUCTION

Introducing mobile cloud computing techniques has empowered numerous applications in people's life in recent years. Involving humans in the cloud computing and wireless connection loops becomes an alternation for information retrieval deriving from observing humans' behaviors and interactivities over various social networks and mobile apps. Moreover, as an emerging technology, cloud computing has spread into countless fields so that many new service deployments are introduced to the public, such as mobile parallel computing and distributed scalable data storage. Penetrations of big data techniques have further enriched the channels of gaining information from the large volume of mobile apps' data across various platforms, domains, and systems. Being one of technical mainstreams has enabled big data to be widely applied in multiple industrial domains as well as explored in recent research.

Chapter 2

LITERATURE SURVEY

K. Gai, L. Qiu, M. Chen, H. Zhao, and M. Qiu. SA-EAST: security aware efficient data transmission for ITS in mobile heterogeneous cloud computing. ACM Transactions on Embedded Computing Systems, 16(2):60, 2017.

The expected advanced network explorations and the growing demand for mobile data sharing and transferring have driven numerous novel applications in *Cyber-Physical Systems* (CPSs), such as *Intelligent Transportation Systems* (ITSs). However, current ITS implementations are restricted by the conflicts between security and communication efficiency. Focusing on this issue, this article proposes a *Security-Aware Efficient Data Sharing and Transferring* (SA-EAST) model, which is designed for securing cloud-based ITS implementations. In applying this approach, we aim to obtain secure real-time multimedia data sharing and transferring. Our experimental evaluation has shown that our proposed model provides an effective performance in securing communications for ITS.

L. Wu, K. Wu, A. Sim, M. Churchill, J. Choi, A. Stathopoulos, C. Chang, and S. Klasky. Towards real-time detection and tracking of spatio-temporal features: Blob-filaments in fusion plasma. IEEE Transactions on Big Data, 2(3), 2016.

A novel algorithm and implementation of real-time identification and tracking of blob-filaments in fusion reactor data is presented. Similar spatio-temporal features are important in many other applications, for example, ignition kernels in combustion and tumor cells in a medical image. This work presents an approach for extracting these features by dividing the overall task into three steps: local identification of feature cells, grouping feature cells into extended feature, and tracking movement of feature through overlapping in space.

S. Yu, W. Zhou, S. Guo, and M. Guo. A feasible IP traceback framework through dynamic deterministic packet marking. IEEE Transactions on Computers, 65(5):1418–1427, 2016.

DDoS attack source traceback is an open and challenging problem. Deterministic packet marking (DPM) is a simple and effective traceback mechanism, but the current DPM based traceback schemes are not practical due to their scalability constraint. We noticed a factor that only a limited number of computers and routers are involved in an attack session. Therefore, we only need to mark these involved nodes for traceback purpose, rather than marking every node of the Internet as the existing schemes doing. Based on this finding, we propose a novel marking on demand (MOD) traceback scheme based on the DPM mechanism. In order to traceback to involved attack source, what we need to do is to mark these involved ingress routers using the traditional DPM strategy. Similar to existing schemes, we require participated routers to install a traffic monitor. When a monitor notices a surge of suspicious network flows, it will request a unique mark from a globally shared MOD server, and mark the suspicious flows with the unique marks. At the same time, the MOD server records the information of the marks and their related requesting IP addresses. Once a DDoS attack is confirmed, the victim can obtain the attack sources by requesting the MOD server with the marks extracted from attack packets. Moreover, we use the marking space in a round-robin style, which essentially addresses the scalability problem of the existing DPM based traceback schemes. We establish a mathematical model for the proposed traceback scheme, and thoroughly analyze the system. Theoretical analysis and extensive real-world data experiments demonstrate that the proposed traceback method is feasible and effective.

Chapter 3

SYSTEM REQUIREMENT SPECIFICATION

3.1 HARDWARE REQUIREMENT

• **Processor**	:	733
• **Keyboard**	:	104 Keys
• **Floppy Drive**	:	1.44 MB MHz Pentium III
• **RAM**	:	128 MB
• **Hard Disk**	:	10GB
• **Monitor**	:	14” VGA COLOR
• **Mouse**	:	Logitech Serial Mouse
• **Disk Space**	:	1 GB

3.2 SOFTWARE REQUIREMENTS

• **Operating System**	:	Win 2000/ XP
• **Technologies used**	:	Java, Servlets, JSP, JDBC
• **JDK**	:	Version 1.4
• **Database**	:	My SQL 5.0

Chapter 4
PROPOSED SYSTEM

In the proposed system we concentrate on privacy and propose a novel data encryption approach, which is called Dynamic Data Encryption Strategy (D2ES). Our proposed approach aims to selectively encrypt data and use privacy classification methods under timing constraints. This approach is designed to maximize the privacy protection scope by using a selective encryption strategy within the required execution time requirements.

ADVANTAGES OF THE PROPOSED SYSTEM

- Maximize the privacy protection scope by using a selective encryption strategy within the required execution time requirements.
- Efficiency and Timing constraints.

Chapter 5

SYSTEM DESIGN

System design of the project is given is detail in this chapter.

5.1 LOGICAL DESIGN

Design for WebApps encompasses technical and non-technical activities. The look and feel of content is developed as part of graphic design; the aesthetic layout of the user interface is created as part of interface design; and the technical structure of the WebApp is modeled as part of architectural and navigational design.

This argues that a Web engineer must design an interface so that it answers three primary questions for the end-user:

1. *Where am I?* – The interface should (1) provide an indication of the WebApp has been accessed and (2) inform the user of her location in the content.

2. *What can I do now?* – The interface should always help the user understand his current options- what functions are available, what links are live, what content is relevant.

3. *Where have I been; where am I going?* – The interface must facilitate navigation. Hence it must provide a “map” of where the user has been and what paths may be taken to move elsewhere in the WebApp.

5.2 DESIGN GOALS

– the following are the design goals that are applicable to virtually every WebApp regardless of application domain, size, or complexity.

1. Simplicity
2. Consistency
3. Identity
4. Visual appeal
5. Compatibility.

Design leads to a model that contains the appropriate mix of aesthetics, content, and technology. The mix will vary depending upon the nature of the WebApp, and as a consequence the design activities that are emphasized will also vary.

The activities of the Design process:

1. Interface design-describes the structure and organization of the user interface. Includes a representation of screen layout, a definition of the modes of interaction, and a description of navigation mechanisms. Interface Control mechanisms- to implement navigation options, the designer selects form one of a number of interaction mechanism;

   a. Navigation menus
   b. Graphic icons
   c. Graphic images

   Interface Design work flow- the work flow begins with the identification of user, task, and environmental requirements. Once user tasks have been identified, user scenarios are created and analyzed to define a set of interface objects and actions.

2. Aesthetic design-also called graphic design, describes the “look and feel” of the WebApp. Includes color schemes, geometric layout. Text size, font and placement, the use of graphics, and related aesthetic decisions.

3. Content design-defines the layout, structure, and outline for all content that is presented as part of the WebApp. Establishes the relationships between content objects.

4. Navigation design-represents the navigational flow between contents objects and for all WebApp functions.

5. Architecture design-identifies the overall hypermedia structure for the WebApp. Architecture design is tied to the goals establish for a WebApp, the content to be presented, the users who will visit, and the navigation philosophy that has been established.

   a. Content architecture, focuses on the manner in which content objects and structured for presentation and navigation.

   b. WebApp architecture, addresses the manner in which the application is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. WebApp architecture is defined within the context of the development environment in which the application is to be implemented.

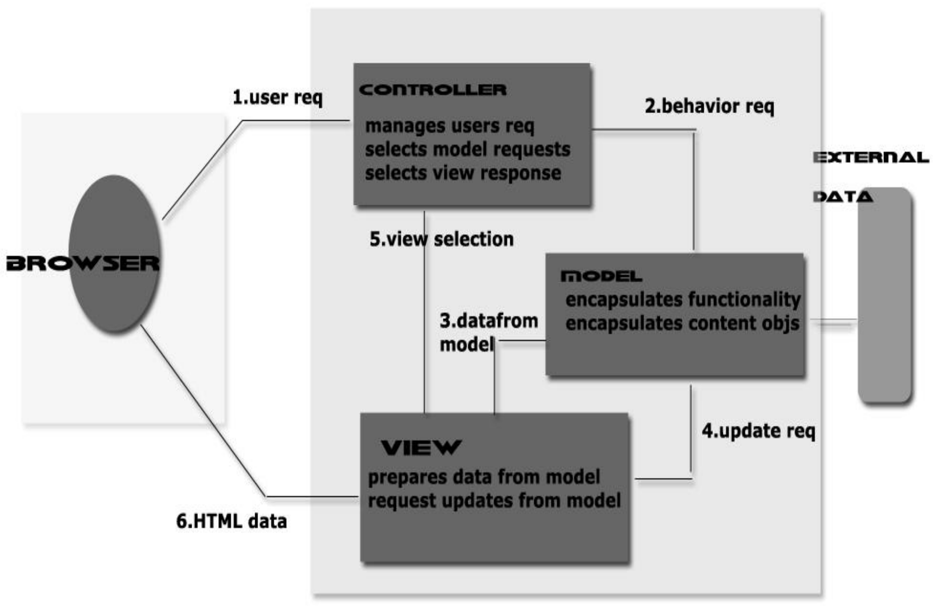


Fig 5. 1 J2EE uses MVC Architecture

6. Component design-develops the detailed processing logic required to implement functional components.

5.3 SOFTWARE ARCHITECTURE

The System architecture is shown below.

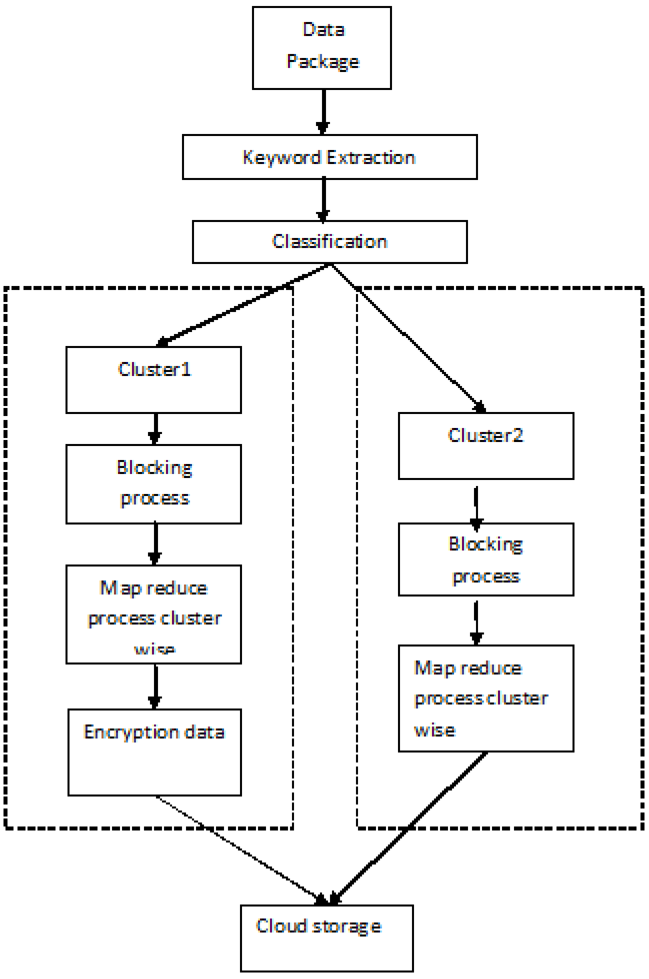


Fig 5. 2 System Architecture

5.4 CLASSES DESIGNED FOR THE SYSTEM

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

The class diagram is shown below.

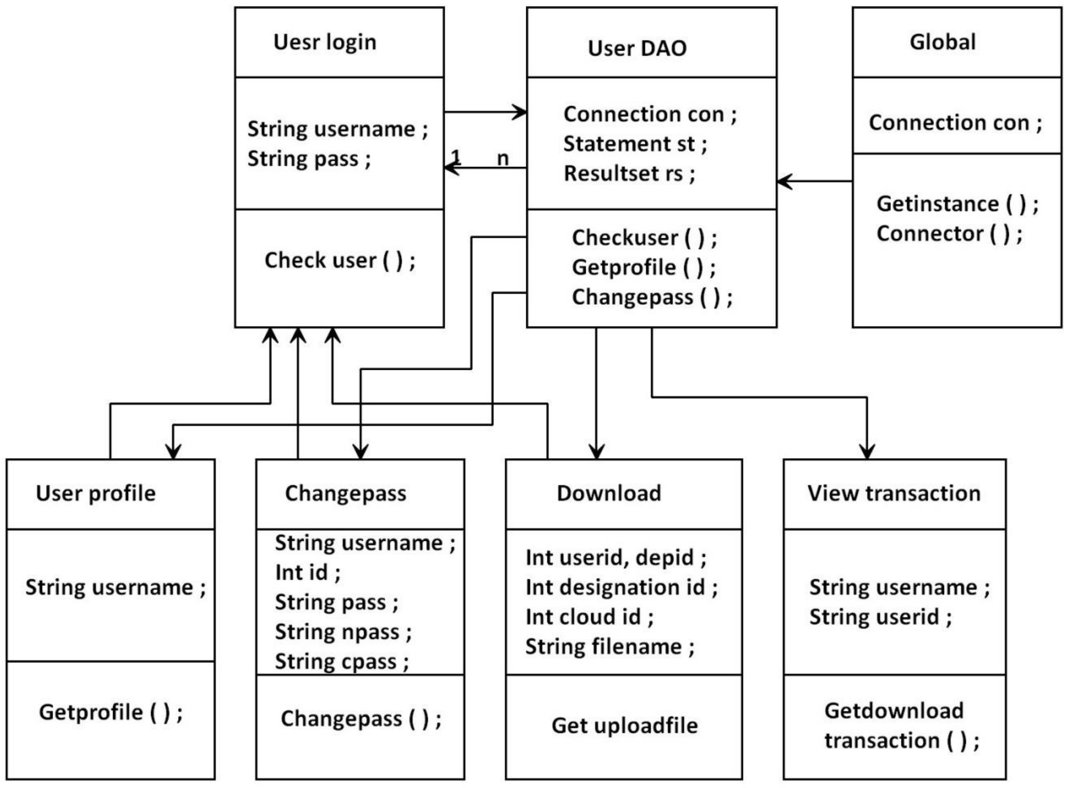


Fig 5. 3 Class Diagram For User

5.5 USE CASE DIAGRAM OF THE SYSTEM

A use case diagram is a type of behavioral diagram created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

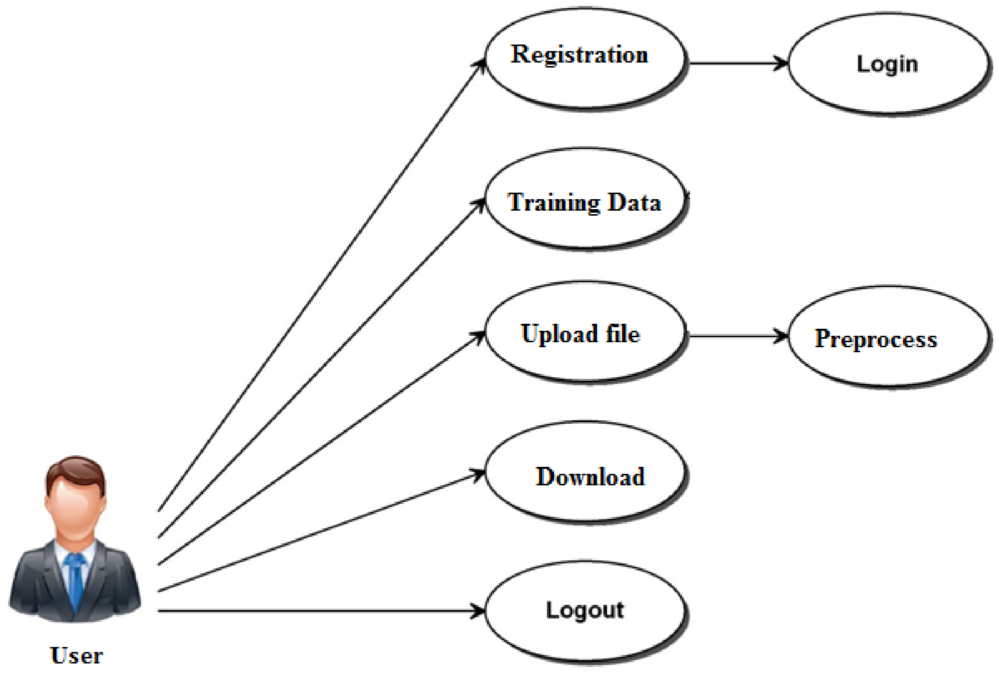


Fig 5. 4 : Use Case Diagram for User

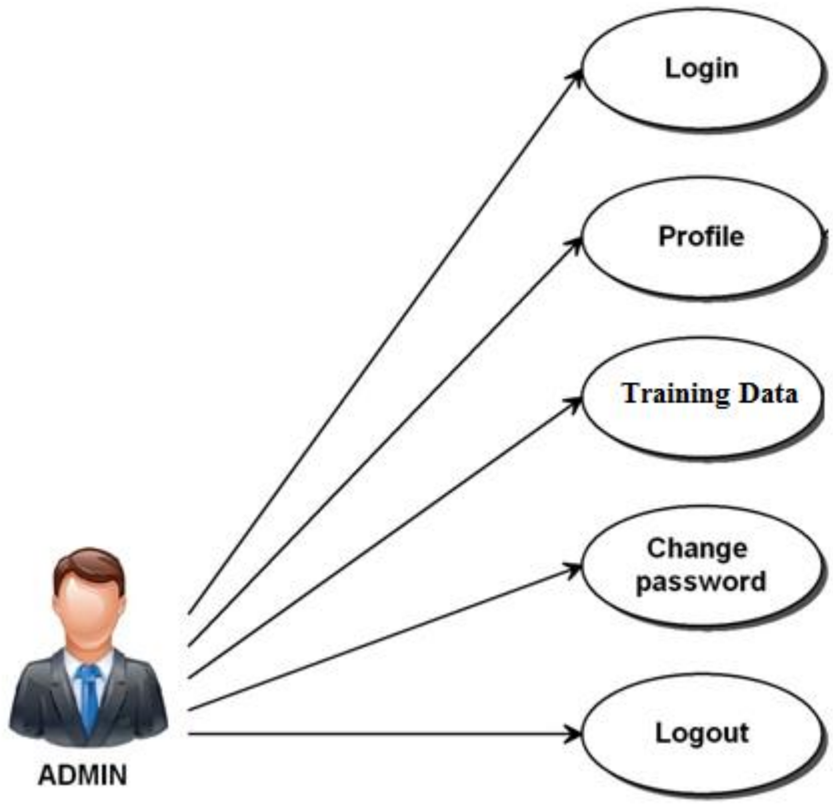


Fig 5. 5: Use Case Diagram for Admin

There are two users

1. Admin
2. User

5.6 DATA FLOW DIAGRAM OF THE SYSTEM

The input, output and the process flow in the system is given in this section.

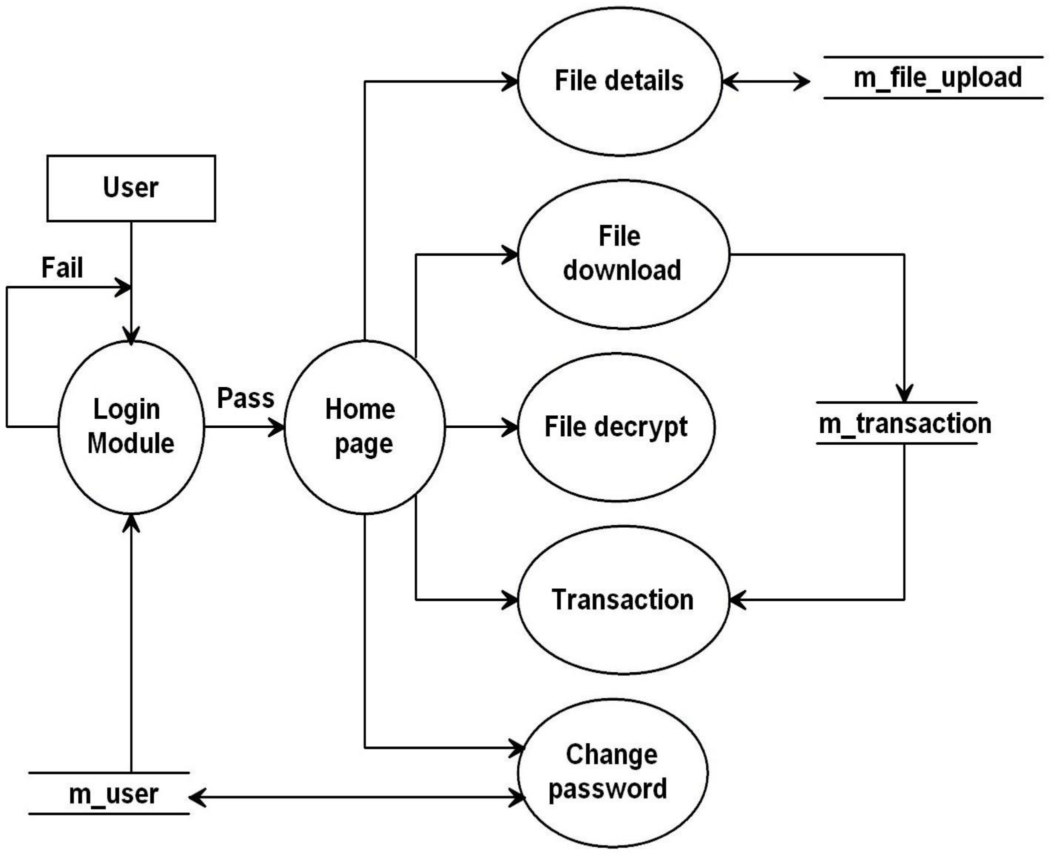


Fig 5. 6 : Dataflow Diagram of User Session

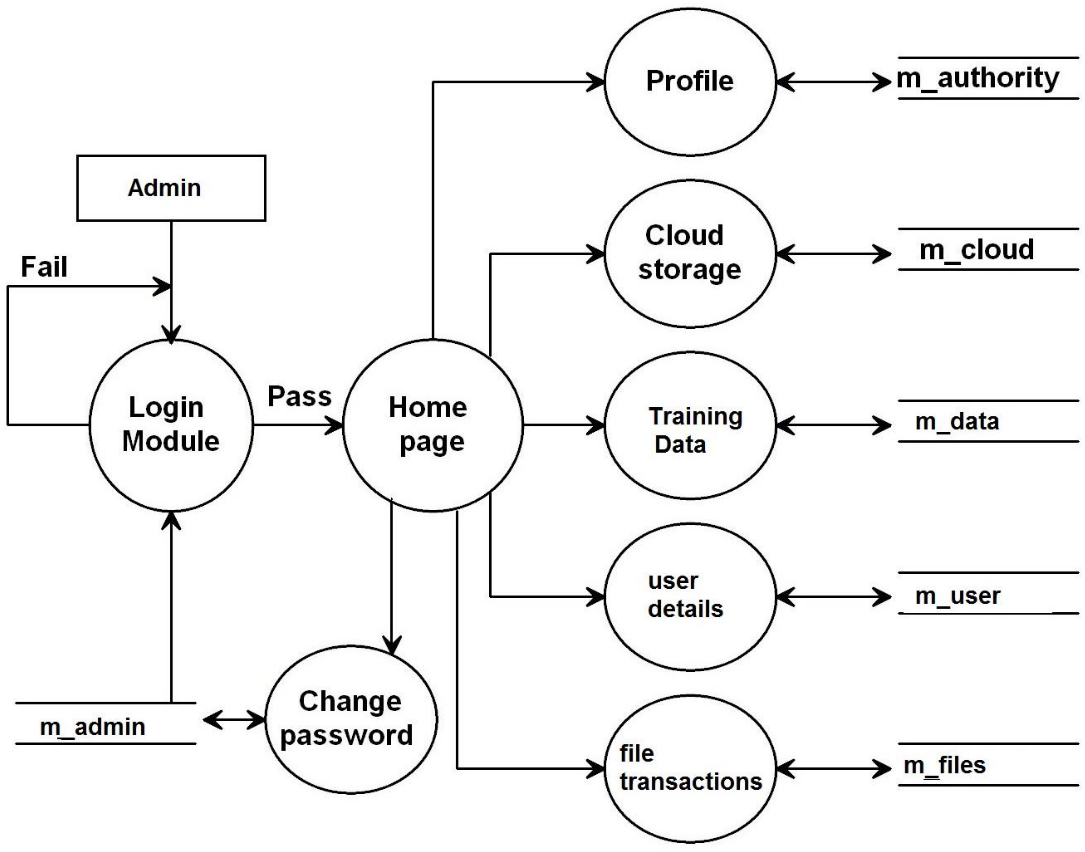


Fig 5. 7 Dataflow Diagram for Admin

5.7 SEQUENCE DIAGRAM OF THE SYSTEM

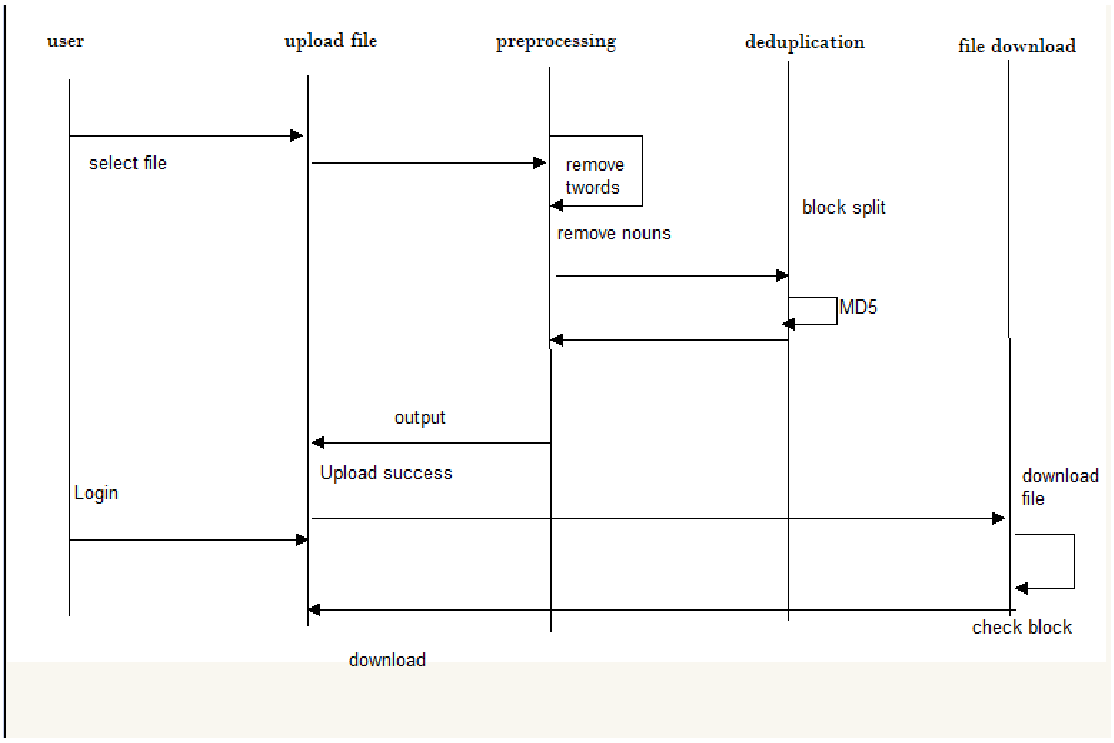


Fig 5. 8: Sequence Diagram for the System

5.8 CONTEXT ANALYSIS DIAGRAM

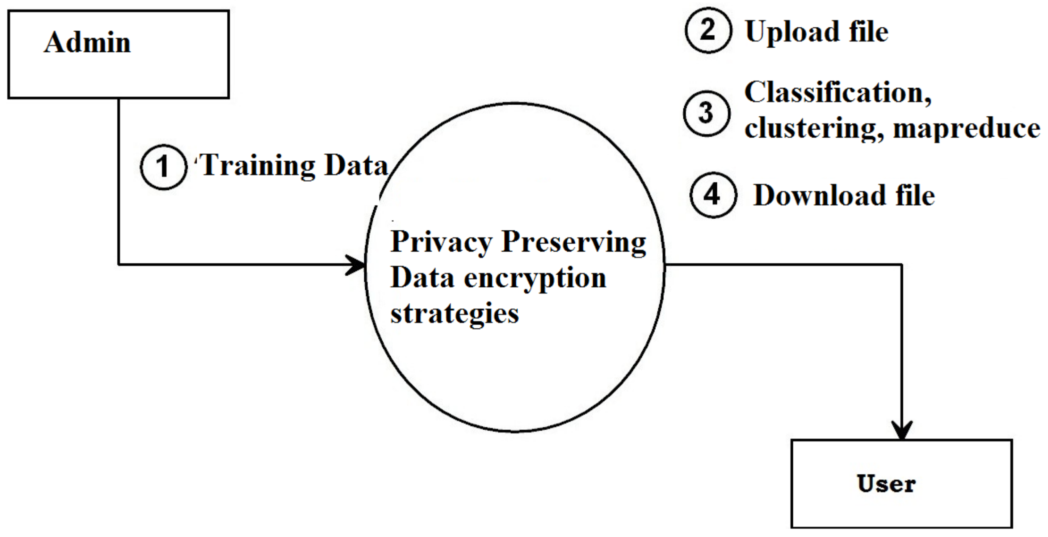

Admin
1 Training Data
2 Upload file
3 Classification, clustering, mapreduce
4 Download file
Privacy Preserving Data encryption strategies
User

Fig 5. 9: Context Analysis Diagram

CHAPTER 6

IMPLEMENTATION

The implementation details of the project are given in detail in this section.

6.1 SAMPLE CODING

```
package com.sentiment;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.ResultSet;
import java.util.ArrayList;
import javax.servlet.RequestDispatcher;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import com.dataa.Sentimentprocessexicution;
import java.util.HashMap;
import java.util.Map;
public class Sentimentprocess extends HttpServlet
{
        public void doGet(HttpServletRequest request,HttpServletResponse
        response)throws IOException
        {
                String college="";
                ArrayList<String> data=new ArrayList<String>();
                String sensitivetype="";
```

```
boolean flag77=false;
boolean ff=false;
int positivecount11=0;
int negativecount11=0;
int undecidablecount11=0;
int neutralcount11=0;
int totalcount =0;
boolean finalcount=false;
try
{
System.out.println("its came Sentimentprocess fetch tweets");
int no=Integer.parseInt(request.getParameter("no"));
System.out.println("action is >>>>>>>>>>>>>"+no);
if(no==1)
{
        RequestDispatcher
        rd=request.getRequestDispatcher("/jsp/sentiment.jsp");
        rd.forward(request, response);
}

if(no==2)
{
        HttpSession session = request.getSession(false);
        if ( session.getAttribute( "waitPage" ) == null )
        {
        session.setAttribute( "waitPage", Boolean.TRUE );
        PrintWriter out1 = response.getWriter();
        out1.println( "<html><head>" );
        out1.println( "<title>Please Wait...</title>" );
        out1.println( "<meta http-equiv=\"Refresh\" content=\"0\">" );
        out1.println( "</head><body bgcolor='#2E8B57'>" );
        out1.println( "<br><br><br>" );
        out1.print( "<center><img src='animated.gif' width='400'
        height='400'></img><br><br>");
        out1.println("<font color='#fedcba' size='5'>");
        out1.println( "Please Do not press Back or Refresh
        button.......<br>  " );
        out1.println("<font color='#fedcba' size='5'>");
        out1.println( "Sentiment Detection  is going on " );
        out1.println("<font color='#fedcba' size='5'>");
        out1.println( "Please wait....</h1></center");
        out1.close();
}
else
{
        session.removeAttribute( "waitPage" );
        college=request.getParameter("select");
        System.out.println("college is
        >>>>>>>>>>>>>>>>>>>>>>>>"+college);
        HashMap<Integer, ArrayList<String>> map_tclaim=new
        HashMap<Integer, ArrayList<String>>();
        map_tclaim=AdminDAO.getm_tweets(college);
        map_tclaim.size();
        System.out.println(">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
        >size"+map_tclaim.size());
        System.out.println("entry set
        is>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"+map_tclai
        m.entrySet());
```

```
if(map_tclaim.size()>0)
{

        for(Map.Entry m4:map_tclaim.entrySet())
        {
        // System.out.println("entry set
        is>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"+ma
        p_tclaim.entrySet());
        m4.getKey();
        String values = m4.getValue().toString();
        String first=values.replace("[", "").replace("]", "");
        String a[]=first.split("~~");
        String part1=a[0];
        System.out.println("Column 1
        is>>>>>>>>>>>>>>>>>>>>"+part1);
        String part2=a[1];
        System.out.println("Column 2
        is>>>>>>>>>>>>>>>>>>>>"+part2);
        /*   part5= Temp.removehttplink(part5);*/
        part2= Temp.removeUrl(part2);
        System.out.println("String part2
        is>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>"+part2);
        data=Preprocessing1.filter(part2);
        StringBuffer buf = null;
        String str=""+data;
        data.add(str);
        for(int ii=0;ii<data.size();ii++)
        {
        buf=new StringBuffer();
```

```
                    buf.append(data.get(ii));
                }
                String s=buf.toString();
                s=s.replace("[", "");
                s=s.replace("]", "");
                ff= AdminDAO.storefiltereddata(part1, college,s);
            }
        boolean dd=    Sentimentprocessexicution.senti(college);
        System.out.println("dddddddddd"+dd);
        if(dd)
{
        String poss="positive";
        String negg="negative";
        String neuu="neutral";

        String undecidable="undecidable";
        positivecount11 = PoolingDAO.getpositivecount(college,poss);
        System.out.println("positivecount is
        >>>>>>>>>>>>>>>>>>>>"+positivecount11);
        negativecount11 = PoolingDAO.getnegativecount(college,negg);
        System.out.println("negativecount is
        >>>>>>>>>>>>>>>>>>>>"+negativecount11);
        neutralcount11 = PoolingDAO.getneuralcount(college,neuu);
        System.out.println("nuetralcount is
        >>>>>>>>>>>>>>>>>>>>"+neutralcount11);



        undecidablecount11 =
        PoolingDAO.getundecidablecount(college,undecidable);
        System.out.println("undecidablecount is
        >>>>>>>>>>>>>>>>>>>>"+undecidablecount11);

        totalcount = PoolingDAO.gettotalcount(college);
        System.out.println("totalcount is >>>>>>>>>>>>>>>>>>>>"+totalcount);
        finalcount=
        AdminDAO.insertfinalcount(college,positivecount11,negativecount11,neutral
        count11,undecidablecount11,totalcount);
}
}
}
}
if(finalcount)
{
        ResultSet rs=LoginDao.gettweets(college);
        request.setAttribute("rs",rs);

        RequestDispatcher
        rd=request.getRequestDispatcher("/jsp/viewtsentimenttweets.jsp?no=2&pmar
        k="+positivecount11+"&negmrk="+negativecount11+"&neumark="+neutralc
        ount11+"&unmark="+undecidablecount11+"&totcount="+totalcount+"");
        rd.forward(request, response);
}
}
catch (Exception e) {
// TODO: handle exception
}
}
```

6.2 DNA IMPLEMENTATION

Phase1: Encryption of Secret Data

Algorithm for Encryption:

Step one: Convert binary data to DNA sequences.

A=00,

T=01,

C=10, and

G=11.

Step two: Complementary pair rule.

Complementary pair rule is a unique equivalent pair which is assigned to every nucleotides base pair.

Example:

Complementary rule: ((AC) (CG) (GT) (TA))

DNA strand: AATGCT

Applying complementary rule on DNA strand: CCATGA.

Step three: Representing DNA sequences as numeric data.

We extract the index of each couple nucleotides in DNA reference sequence, numerically.

Example:

Assume the reference sequence to be $CT_1GA_2TC_3CC_4GC_5AT_6TT_7$.

Then the numerical representation will be 040602.

In this implementation, the admin assigns a unique DNA sequence and a unique key to every user which is later used to create the DNA reference sequence for the user using rand() function and hash set constructs.

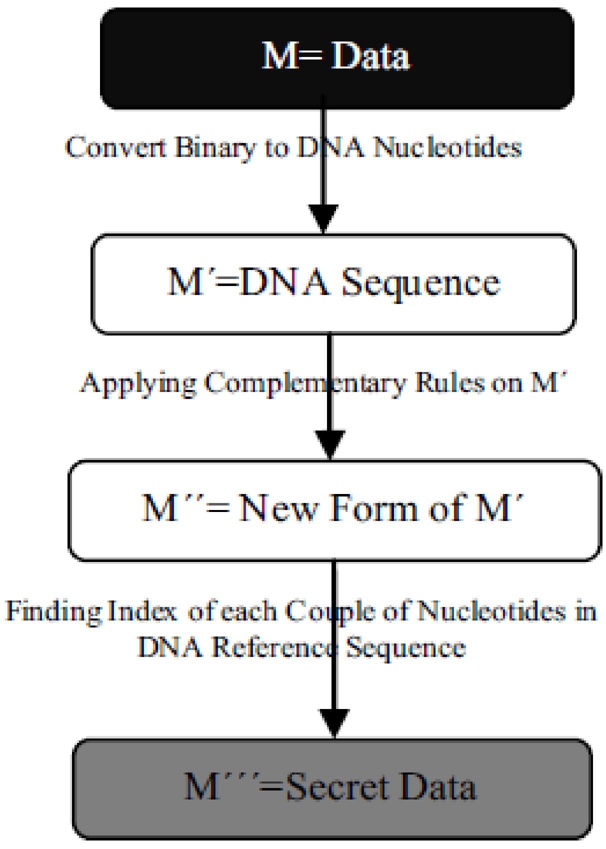

M= Data
Convert Binary to DNA Nucleotides
M´=DNA Sequence
Applying Complementary Rules on M´
M´´= New Form of M´
Finding Index of each Couple of Nucleotides in DNA Reference Sequence
M´´´=Secret Data

Fig 6. 1 Encryption Process

There is an original data M which the client decides to upload via a network to cloud computing environments. So, there are three sub-phases to provide the final form of M which is M´´´ and upload it to cloud. The data M is read as integer and converted to binary form.

In order to convert binary data into amino acids as a DNA sequence, the base pairing rules must be used. Synthesizing nucleotides in real environment (biology) is done in constant rules.

Example:

Assume original data M=01000001(A) should be uploaded to the cloud

- DNA Reference Sequence:

  [TG, TA, AT, GC, CT, GA, CA, AC, AA, GT, CG, AG, CC, TT, TC, GG]

  [TG$_{\mathbf{00}}$,TA$_{\mathbf{01}}$,AT$_{\mathbf{02}}$,GC$_{\mathbf{03}}$,CT$_{\mathbf{04}}$,GA$_{\mathbf{05}}$,CA$_{\mathbf{06}}$,AC$_{\mathbf{07}}$,AA$_{\mathbf{08}}$,GT$_{\mathbf{09}}$,CG$_{\mathbf{10}}$,AG$_{\mathbf{11}}$,CC$_{\mathbf{12}}$,TT$_{\mathbf{13}}$,TC$_{\mathbf{14}}$,GG$_{\mathbf{15}}$]

- M=01000001 (Original data).
- Sub-phase1 (Base pairing rule)

  (A= 00, T= 01, C= 10, G= 11): M´= TAAT
- Sub-phashe2 (Applying complimentary rule)

  ((AC) (CG) (GT) (TA)): M´´= ACCA
- Sub-phase3 (Indexes):M´´´= 0706**(Encrypted data**)

Phase2: Extracting Original data

Client2 takes the secret data in form of some numbers. For the purpose of extracting the original data from DNA reference sequence, phase two with its sub phases will extract the original data, correctly depicted in figure 7.3.1.

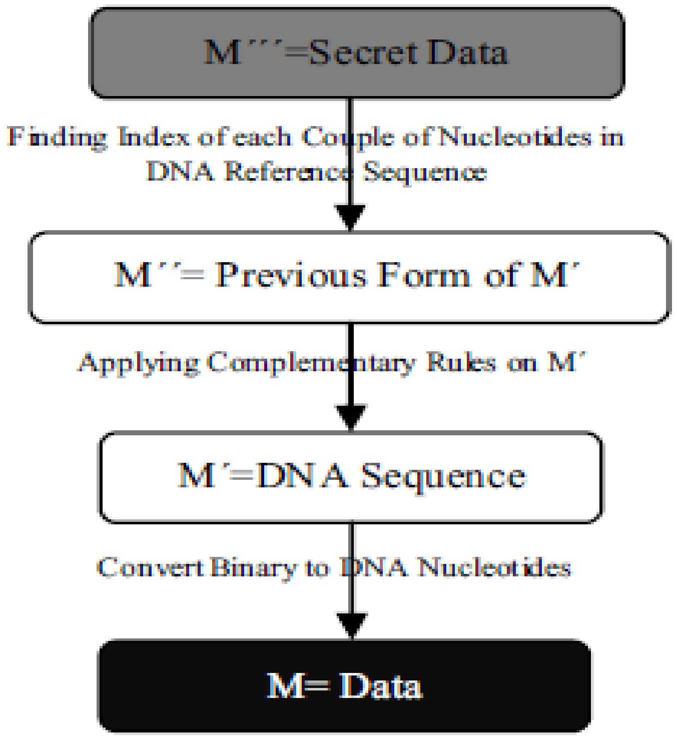

M´´´=Secret Data
Finding Index of each Couple of Nucleotides in DNA Reference Sequence
M´´= Previous Form of M´
Applying Complementary Rules on M´
M´=DNA Sequence
Convert Binary to DNA Nucleotides
M= Data

Fig 6. 2: Decryption Process

When the user is uploading a file from his local machine to web server, it has to be redirected to DNA Encryption Service which is running in cloud server 1 and Cloud server 1 will encrypt the file and the encrypted file has to send to cloud server 2 using web service concept which will store in cloud server 2.

Algorithm for Decryption:

Step One: Convert numeric data to DNA sequences.

We extract the couple nucleotides in DNA reference sequence according to the index read from the file.

Step two: Complementary pair rule.

Complementary pair rule is a unique equivalent pair which is assigned to every nucleotides base pair.

Step three: Convert DNA sequences to binary data.

Example:

DNA Reference Sequence:

[TG$_{00}$,TA$_{01}$,AT$_{02}$,GC$_{03}$,CT$_{04}$,GA$_{05}$,CA$_{06}$,AC$_{07}$,AA$_{08}$,GT$_{09}$,CG$_{10}$,AG$_{11}$,CC$_{12}$,TT$_{13}$,TC$_{14}$, GG$_{15}$]

[TG, TA, AT, GC, CT, GA, CA, AC, AA, GT, CG, AG, CC, TT, TC, GG]

- M´´´=0706 (Input)
- By referring the DNA sequence:

  Sub-phase1 (Indexes): M´´= ACCA.
- By using Complementary rule:

  Sub-phase2 ((AC) (CG) (GT) (TA)):M´= TAAT
- By using Base Pair Rule:

  Sub-phase3 (A= 00, T= 01, C= 10, G= 11):

  M=01000001 (A)**(Output)**

When the user is downloading a file from the web server the corresponding file has to fetch from cloud server 2 and it will send to DNA decryption service which is running in cloud server 1. Once the file is decrypted it will be downloaded to the user machine [1].

6.3 AN OVERVIEW

6.3.1 HTML

HTML, an initialism of Hypertext Markup Language, is the predominant markup language for web pages. It provides a means to describe the structure of text-based information in a document — by denoting certain text as headings, paragraphs, lists, and so on — and to supplement that text with interactive forms, embedded images, and other objects. HTML is written in the form of labels (known as tags), surrounded by angle brackets. HTML can also describe, to some degree, the appearance and semantics of a document, and can include embedded scripting language code which can affect the behavior of web browsers and other HTML processors.

HTML is also often used to refer to content of the MIME type text/html or even more broadly as a generic term for HTML whether in its XML-descended form (such as XHTML 1.0 and later) or its form descended directly from SGML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

Basic HTML Tags:

<! -- -->	specifies comments
<A>……….</A>	Creates hypertext links
<B>……….</B>	Formats text as bold
<BIG>……….</BIG>	Formats text in large font.
<BODY>…</BODY> document	Contains all tags and text in the HTML
<CENTER>...</CENTER>	Creates text
<DD>…</DD>	Definition of a term
<DL>...</DL>	Creates definition list
<FONT>…</FONT>	Formats text with a particular font
<FORM>...</FORM>	Encloses a fill-out form
<FRAME>...</FRAME>	Defines a particular frame in a set of frames
<H#>…</H#>	Creates heading of different levels(1 – 6)
<HEAD>...</HEAD>	Contains tags that specify information about a document
<HR>...</HR>	Creates a horizontal rule
<HTML>…</HTML>	Contains all other HTML tags
<META>...</META>	Provides meta-information about a document
<SCRIPT>…</SCRIPT>	Contains client-side or server-side script
<TABLE>…</TABLE>	Creates a table
<TD>…</TD>	Indicates table data in a table
<TR>…</TR>	Designates a table row
<TH>…</TH>	Creates a heading in a table

Attributes

The attributes of an element are name-value pairs, separated by "=", and written within the start label of an element, after the element's name. The value should be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML).Leaving attribute values unquoted is considered unsafe.

Most elements take any of several common attributes: id, class, style and title. Most also take language-related attributes: lang and dir.

The id attribute provides a document-wide unique identifier for an element. This can be used by style sheets to provide presentational properties, by browsers to focus attention on the specific element or by scripts to alter the contents or presentation of an element. The class attribute provides a way of classifying similar elements for presentation purposes. For example, an HTML document (or a set of documents) may use the designation class="notation" to indicate that all elements with this class value are all subordinate to the main text of the document (or documents). Such notation classes of elements might be gathered together and presented as footnotes on a page, rather than appearing in the place where they appear in the source HTML.

An author may use the style non-attributal codes presentational properties to a particular element. It is considered better practice to use an element’s son- id page and select the element with a style sheet, though sometimes this can be too cumbersome for a simple ad hoc application of styled properties. The title is used to attach sub textual explanation to an element. In most browsers this title attribute is displayed as what is often referred to as a tool tip. The generic inline span element can be used to demonstrate these various non-attributes.

Advantages

- A HTML document is small and hence easy to send over the net.
- It is small because it does not include formatted information.
- HTML is platform independent.
- HTML tags are not case-sensitive.

6.3.2 JAVASCRIPT

JavaScript is a script-based programming language that was developed by Netscape Communication Corporation. JavaScript was originally called Live Script and renamed as JavaScript to indicate its relationship with Java. JavaScript supports the development of both client and server components of Web-based applications. On the client side, it can be used to write programs that are executed by a Web browser within the context of a Web page. On the server side, it can be used to write Web server programs that can process information submitted by a Web browser and then update the browser's display accordingly

Even though JavaScript supports both client and server Web programming, we prefer JavaScript at Client side programming since most of the browsers supports it. JavaScript is almost as easy to learn as HTML, and JavaScript statements can be included in HTML documents by enclosing the statements between a pair of scripting tags

```
<Script> ……… </Script>
        <Script Language = "JavaScript">
                JavaScript statements
        </Script>
```

Here are a few things we can do with JavaScript

- Validate the contents of a form and make calculations.

- Add scrolling or changing messages to the Browser's status line.
- Animate images or rotate images that change when we move the mouse over them.
- Detect the browser in use and display different content for different browsers.
- Detect installed plug-ins and notify the user if a plug-in is required.
- We can do much more with JavaScript, including creating entire application.

JavaScript and Java are entirely different languages. A few of the most glaring differences are:

- Java applets are generally displayed in a box within the web document; JavaScript can affect any part of the Web document itself.
- While JavaScript is best suited to simple applications and adding interactive features to Web pages; Java can be used for incredibly complex applications.

There are many other differences but the important thing to remember is that JavaScript and Java are separate languages. They are both useful for different things; in fact they can be used together to combine their advantages.

- ➢ JavaScript can be used for Sever-side and Client-side scripting.
- ➢ It is more flexible than VBScript.
- ➢ JavaScript is the default scripting languages at Client-side since all the browsers supports it.

6.3.3 JAVA TECHNOLOGY

Initially the language was called as “*oak*” but it was renamed as “*Java*” in 1995. The primary motivation of this language was the need for a platform-independent (i.e., architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language.
- Java is cohesive and consistent.
- Except for those constraints imposed by the Internet environment, Java gives the programmer, full control.

- Finally, Java is to Internet programming where C was to system programming.

IMPORTANCE OF JAVA TO THE INTERNET

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the Server and the Personal computer. They are: Passive information and Dynamic active programs. The Dynamic, Self-executing programs cause serious problems in the areas of Security and probability. But, Java addresses those concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

JAVA CAN BE USED TO CREATE TWO TYPES OF PROGRAMS

Applications and Applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++. Java’s ability to create Applets makes it important. An Applet is an application designed to be transmitted over the Internet and executed by a Java –compatible web browser. An applet is actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can react to the user input and dynamically change.

FEATURES OF JAVA SECURITY

Every time you that you download a "normal" program you are risking a viral infection. Prior to java, most users did not download executable programs frequently, and those who did scan them for viruses prior to execution. Most users still worried about the possibility of infecting their systems with a virus. In addition, another type of malicious program exists that must be guarded against. This type of program can gather private information, such as credit card numbers, bank account balances, and passwords. Java answers both these concerns by providing a "firewall" between a network application and your computer.

PORTABILITY

For programs to be dynamically downloaded to all the various types of platforms connected to the internet, some means of generating portable executable code is needed .as you will see, the same mechanism that helps ensure security also helps create portability. Indeed, java's solution to these two problems is both elegant and efficient.

THE BYTE CODE

The key that allows the Java to solve the security and portability problems is that the output of Java compiler is Byte code. Byte code is a highly optimized set of instructions designed to be executed by the Java run-time system, which is called the Java Virtual Machine (JVM). That is, in its standard form, the JVM is an interpreter for byte code.

Translating a Java program into byte code helps makes it much easier to run a program in a wide variety of environments. The reason is, once the run-time package exists for a given system, any Java program can run on it.

Java Virtual Machine(JVM)

Beyond the language, there is the Java virtual machine. The Java virtual machine is an important element of the Java technology. The virtual machine can be embedded within a web browser or an operating system. Once a piece of Java code is loaded onto a machine, it is verified. As part of the loading process, a class loader is invoked and does byte code verification makes sure that the code that's has been generated by the compiler will not corrupt the machine that it's loaded on. Byte code verification takes place at the end of the compilation process to make sure that is all accurate and correct. So byte code verification is integral to the compiling and executing of Java code.

OVERALL DESCRIPTION

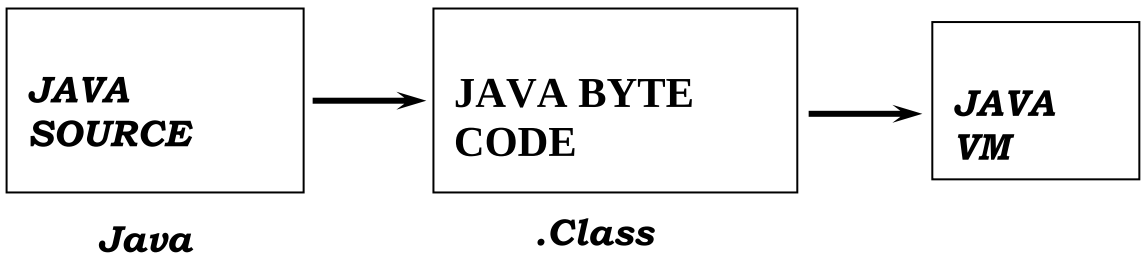


Fig 6. 3: Development Process

Java programming uses to produce byte codes and executes them. The first box indicates that the Java source code is located in a. Java file that is processed with a Java compiler called javac. The Java compiler produces a file called a. class file, which contains the byte code. The .Class file is then loaded across the network or loaded locally on your machine into the execution environment is the Java virtual machine, which interprets and executes the byte code.

JAVA ARCHITECTURE

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

COMPILATION OF CODE

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to execute the byte code. The JVM is created for overcoming the issue of portability. The code is written and compiled for one machine and interpreted on all machines. This machine is called Java Virtual Machine.

6.3.4 INTRODUCTION TO SERVLETS

Servlets provide a Java(TM)-based solution used to address the problems currently associated with doing server-side programming, including inextensible scripting solutions, platform-specific APIs, and incomplete interfaces.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server. Servlets are to the server-side what applets are to the client-side -- object byte codes that can be dynamically loaded off the net. They differ from applets in that they are faceless objects (without graphics or a GUI component). They serve as platform-independent, dynamically loadable, plug gable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

What is a Servlet?

Servlets are modules that extend request/response-oriented servers, such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.

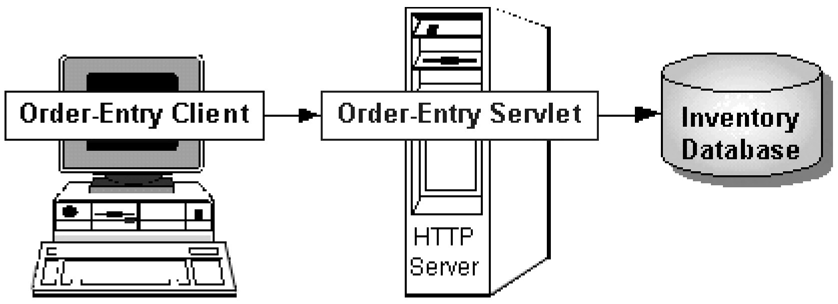


Fig 6. 4: Servlet

Servlets are to servers what applets are to browsers. Unlike applets, however, Servlets have no graphical user interface. Servlets can be embedded in many different servers because the servlet API, which you use to write Servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support the Servlet API.

Use Servlets instead of CGI Scripts

- Servlets are an effective replacement for CGI scripts. They provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform-specific APIs: they are developed with the Java Servlet API, a standard Java extension.

- So use Servlets to handle HTTP client requests. For example, have Servlets process data posted over HTTPS using an HTML form, including purchase order or credit card data.

The Servlet Interface

- The central abstraction in the Servlet API is the Servlet interface. All Servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as HttpServlet.

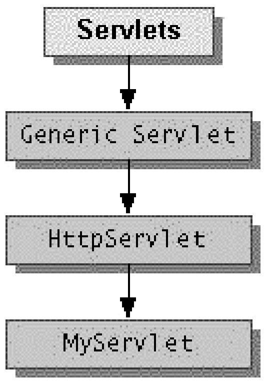


Fig 6. 5: Servlet Interface

- The Servlet interface declares, but does not implement, methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.

Client Interaction

- When a servlet accepts a call from a client, it receives two objects:
- A ServletRequest, which encapsulates the communication from the client to the server.
- A ServletResponse, which encapsulates the communication from the servlet back to the client.
- ServletRequest and ServletResponse are interfaces defined by the javax.servlet package

6.3.5 JAVA SERVER PAGES

Java Server Pages technology lets you put snippets of servlet code directly into a text-based document. A JSP page is a text-based document that contains two types of text: static template data, which can be expressed in any text-based format such as HTML, WML, and XML, and JSP elements, which determine how the page constructs dynamic content.

Java Server Page™ (JSP): An extensible Web technology that uses template data, custom elements, scripting languages, and server-side Java objects to return dynamic content to a client. Typically the template data is HTML or XML elements, and in many cases the client is a Web browser.

According to JSP model1 we can develop the application as,

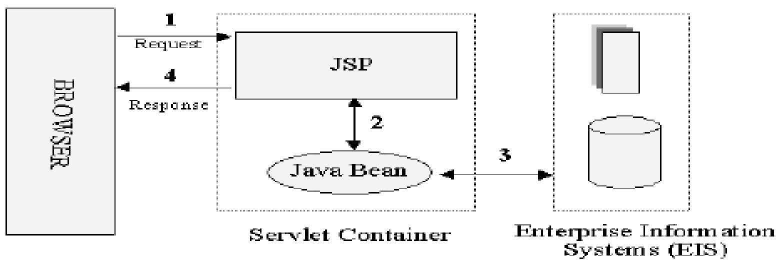


Fig 6. 6: Java Server Pages Model

According to above model the presentation logic has to be implemented in JSP page and the business logic has to be implemented as part of Java bean This model help us in separating the presentation and business logic. For large-scale projects instead of using model1 it is better to use model2 (MVC). Struts framework is based on model 2.

Java Server Pages (JSP) lets you separate the dynamic part of your pages from the static HTML. You simply write the regular HTML in the normal manner, using whatever Web-page-building tools you normally use. You then enclose the code for the dynamic parts in special tags, most of which start with "<%" and end with "%>". For example, here is a section of a JSP page that results in something like "Thanks for ordering Core Web Programming

For URL of

http://host/OrderConfirmation.jsp?title=Core+Web+Programming:

Thanks for ordering

<I><%= request.getParameter("title") %></I>

You normally give your file a .jsp extension, and typically install it in any place you could place a normal Web page. Although what you write often looks more like a regular HTML file than a servlet, behind the scenes, the JSP page just gets converted to a normal servlet, with the static HTML simply being printed to the output stream associated with the servlet's service method. This is normally done the first time the page is requested, and developers can simply request the page themselves when first installing it if they want to be sure that the first real user doesn't get a momentary delay when the JSP page is translated to a servlet and the servlet is compiled and loaded. Note also that many Web servers let you define aliases that so that a URL that appears to reference an HTML file really points to a servlet or JSP page.

Aside from the regular HTML, there are three main types of JSP constructs that you embed in a page: scripting elements, directives, and actions. Scripting elements let you specify Java code that will become part of the resultant servlet, directives let you control the overall structure of the servlet, and actions let you specify existing components that should be used, and otherwise control the behavior of the JSP engine. To simplify the scripting elements, you have access to a number of predefined variables such as request in the snippet above.

6.3.6 J2EE PLATFORM OVERVIEW

The J2EE platform is designed to provide server-side and client-side support for developing distributed, multi-tier applications. Such applications are typically configured as a client tier to provide the user interface, one or more middle-tier modules that provide client services and business logic for an application, and back-end enterprise information systems providing data management.

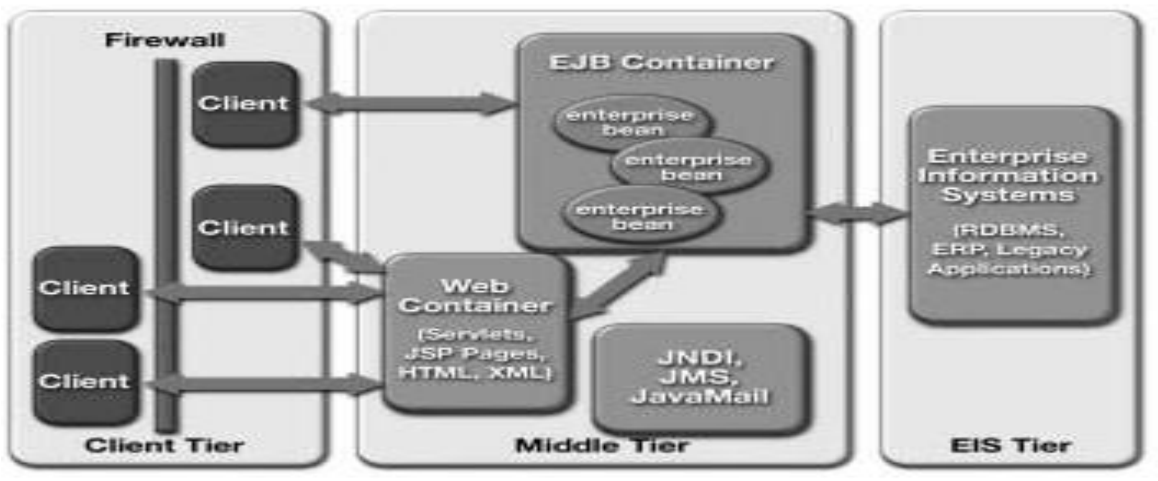


Fig 6. 7: J2EE Platform

Multitier Model

The J2EE platform provides a multi-tier distributed application model. This means that the various parts of an application can run on different devices. The J2EE architecture defines a client tier, a middle tier (consisting of one or more sub-tier), and a back-end tier. The client tier supports a variety of client types, both outside and inside of corporate firewalls. The middle tier supports client services through Web containers in the Web tier and supports business logic component services through JavaBeans TM.

Container-Based Component Management

Central to the J2EE component-based development model is the notion of containers. Containers are standardized runtime environments that provide specific services to components. Components can expect these services to be available on any J2EE platform from any vendor. For example, all J2EE Web containers provide runtime support for responding to client requests, performing request-time processing (such as invoking JSP pages or servlet behavior), and returning results to the client. In addition, they provide APIs to support user session management. All WEB containers provide automated support for transaction and life cycle management of WEB components, as well as bean lookup and other services. Containers also provide standardized access to enterprise information systems; for example, providing access to relational data through the JDBC API. In addition, containers provide a mechanism for selecting application behaviors at assembly or deployment time. Through the use of deployment descriptors (XML files that specify component and container behavior), components can be configured to a specific container's environment when deployed, rather than in component code. Features that can be configured at deployment time include security checks, transaction control, and other management responsibilities.

While the J2EE specification defines the component containers that a platform implementation must support, it doesn't specify or restrict the containers' configurations. Thus, both container types can run on a single platform, Web containers can live on one platform and WEB containers on another or a J2EE platform can be made up of multiple containers on multiple platforms.

Support for Client Components

The J2EE client tier provides support for a variety of client types, both within the enterprise firewall and outside. Clients can be offered through Web browsers by using plain HTML pages, HTML generated dynamically by Java Server PagesTM.

J2EE Application Scenarios

The J2EE specifications encourage architectural diversity. The J2EE specifications and technologies make few assumptions about the details of API implementations. The application-level decisions and choices are ultimately a trade-off between functional richness and complexity. The J2EE programming model is flexible enough for applications that support a variety of client types, with both the Web container and WEB container as optional.

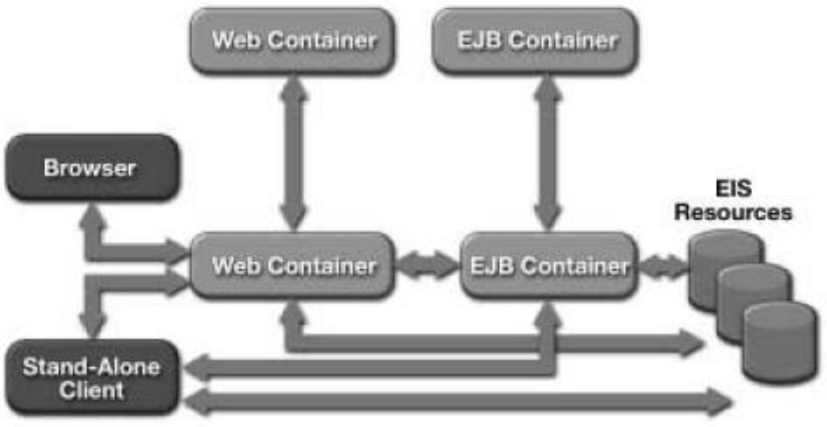


Fig 6. 8: J2EE Application Structure

The following enterprise requirements heavily influenced the choices made in developing the sample application:

- The need to make rapid and frequent changes to the "*look and feel*" of the application.
- The need to partition the application along the lines of presentation and business logic so as to increase modularity
- The need to simplify the process of assigning suitably trained human resources to accomplish the development task such that work can proceed along relatively independent but cooperating tracks

The need to have developers familiar with back-office applications unburdened from GUI and graphic design work, for which they may not be ideally qualified.

The need to have the necessary vocabulary to communicate the business logic to teams concerned with human factors and the aesthetics of the application.

The ability to assemble back-office applications using components from a variety of sources, including off-the-shelf business logic components .

The ability to deploy transactional components across multiple hardware and software platforms independently of the underlying database technology.
The ability to externalize internal data without having to make many assumptions about the consumer of the data and to accomplish this in a loosely coupled manner Clearly, relaxing any or all of these requirements would influence some of the application-level decisions and choices that a designer would make. Although it is reasonable to speak of "throw-away" presentation logic (that is, applications with a look and feel that ages rapidly), there is still significant inertia associated with business logic. This is even truer in the case of database schemas and data in general. It is fair to say that as one moves further away from EIS resources, the volatility of the application code increases dramatically; that is, the code's "shelf life" drops significantly.

6.3.7 MYSQL

Introduction

MySql is a relational database management system, which organizes data in the form of tables. MySQL is one of many databases servers based on RDBMS model, which manages a seer of data that attends three specific things-data structures, data integrity and data manipulation. With MySQL cooperative server technology, we can realize the benefits of open, relational systems for all the applications. MySQL makes efficient use of all systems resources, on all hardware architecture; to deliver unmatched performance, price performance and scalability. Any DBMS to be called as RDBMS has to satisfy Dr.E.F.Codd’s rules.

Distinct Features of MySql:

- **MYSQL IS PORTABLE:**

  The MySQL RDBMS is available on wide range of platforms ranging from PCs to super computers and as a multi user loadable module for Novel NetWare, if you develop application on system you can run the same application on other systems without any modifications.

- **MYSQL IS COMPATIBLE:**

  MySQL commands can be used for communicating with IBM DB2 mainframe RDBMS that is different from MySQL , that is MySQL compatible with DB2 .MySQL RDBMS is a high performance fault tolerant DBMS , which is specially designed for online transaction processing and for handling large database applications.

- **MULTITHREADED SERVER ARCHITECTURE:**

MySQL adaptable multithreaded server architecture delivers scalable high performance for very large number of users on all hardware architecture including symmetric multiprocessors (sumps) and loosely coupled multiprocessors. Performance is achieved by eliminating CPU, I/O, memory and operating system bottlenecks and by optimizing the MySQL DBMS server code to eliminate all internal bottlenecks.

Chapter 7

TESTING

Definition

Unit testing is a development procedure where programmers create tests as they develop software. The tests are simple short tests that test functionally of a particular unit or module of their code, such as a class or function.

Using open source libraries like cunit, oppunit and nun it (for C, C++ and C#) these tests can be automatically run and any problems found quickly. As the tests are developed in parallel with the source unit test demonstrates its correctness.

Validation and System Testing

Validation testing is a concern which overlaps with integration testing. Ensuring that the application fulfils its specification is a major criterion for the construction of an integration test. Validation testing also overlaps to a large extent with ***System Testing***, where the application is tested with respect to its typical working environment. Consequently, for many processes no clear division between validation and system testing can be made. Specific tests which can be performed in either or both stages include the following.

- **Regression Testing:** Where this version of the software is tested with the automated test harness used with previous versions to ensure that the required features of the previous version are skill working in the new version.

- **Recovery Testing:** Where the software is deliberately interrupted in a number of ways off, to ensure that the appropriate techniques for restoring any lost data will function.

- **Security Testing:** Where unauthorized attempts to operate the software, or parts of it, attempted it might also include attempts to obtain access the data, or harm the software installation or even the system software. As with all types of security determined will be able to obtain unauthorized access and the best that can be achieved is to make this process as difficult as possible.

- **Stress Testing:** Where abnormal demands are made upon the software by increasing the rate at which it is asked to accept, or the rate t which it is asked to produce information. More complex tests may attempt to crate very large data sets or cause the soft wares to make excessive demands on the operating system.

- **Performance testing:** Where the performance requirements, if any, are checked. These may include the size of the software when installed, type amount of main memory and/or secondary storage it requires and the demands made of the operating when running with normal limits or the response time.

- **Usability Testing:** The process of usability measurement was introduced in the previous chapter. Even if usability prototypes have been tested whilst the application was constructed, a validation test of the finished product will always be required.

- **Alpha and beta testing:** This is where the software is released to the actual end users. An initial release, the alpha release, might be made to selected users who be expected to report bugs and other detailed observations back to the production team. Once the application changes necessitated by the alpha phase can be made to larger more representative set users, before the final release is made to all users.

The final process should be a **Software audit** where the complete software project is checked to ensure that it meets production management requirements. This ensures that all required documentation has been produced, is in the correct format and is of acceptable quality. The purpose of this review is: firstly to assure the quality of the production process and by implication construction phase commences. A formal hand over from the development team at the end of the audit will mark the transition between the two phases.

- **Integration Testing**:

Integration Testing can proceed in a number of different ways, which can be broadly characterized as top down or bottom up. In top down integration testing the high level control routines are tested first, possibly with the middle level control structures present only as stubs. Subprogram stubs were presented in section2 as incomplete subprograms which are only present to allow the higher. Level control routines to be tested.

Top down testing can proceed in a **depth-first** or a **breadth-first** manner. For depth-first integration each module is tested in increasing detail, replacing more and more levels of detail with actual code rather than stubs. Alternatively breadth-first would processed by refining all the modules at the same level of control throughout the application .in practice a combination of the two techniques would be used. At the initial stages all the modules might be only partly functional, possibly being implemented only to deal with non-erroneous data. These would be tested in breadth-first manner, but over a period of time each would be replaced with successive refinements which were closer to the full functionality. This allows depth-first testing of a module to be performed simultaneously with breadth-first testing of all the modules.

The other major category of integration testing is ***Bottom Up Integration Testing*** where an individual module is tested form a test harness. Once a set of individual module have been tested they are then combined into a collection of modules ,known as builds, which are then tested by a second test harness. This process can continue until the build consists of the entire application. In practice a combination of top down and bottom-up testing would be used. In a large software project being developed by a number of sub-teams, or a smaller project where different modules were built by individuals. The sub teams or individuals would conduct bottom-up testing of the modules which they were constructing before releasing them to an integration team which would assemble them together for top-down testing.

- **Unit Testing:**

Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. The exact scope of a unit is left to interpretation. Supporting test code, sometimes called ***Scaffolding***, may be necessary to support an individual test. This type of testing is driven by the architecture and implementation teams. This focus is also called black-box testing because only the details of the interface are visible to the test. Limits that are global to a unit are tested here.

In the construction industry, scaffolding is a temporary, easy to assemble and disassemble, frame placed around a building to facilitate the construction of the building. The construction workers first build the scaffolding and then the building. Later the scaffolding is removed, exposing the completed building.

similarly, in software testing, one particular test may need some supporting software. This software establishes can a correct evaluation of the test take place. The scaffolding software may establish state and values for data structures as well as providing dummy external functions for the test. Different scaffolding software may be needed form one test to another test. Scaffolding software rarely is considered part of the system.

Sometimes the scaffolding software becomes larger than the system software being tested. Usually the scaffolding software is not of the same quality as the system software and frequently is quite fragile. A small change in test may lead to much larger changes in the scaffolding.

Internal and unit testing can be automated with the help of coverage tools. Analyzes the source code and generated a test that will execute every alternative thread of execution. Typically, the coverage tool is used in a slightly different way. First the coverage tool is used to augment the source by placing information prints after each line of code. Then the testing suite is executed generating an audit trail. This audit trail is analyzed and reports the percent of the total system code executed during the test suite. If the coverage is high and the untested source lines are of low impact to the system's overall quality, then no more additional tests are required.

Chapter 8

INTERPRETATION OF RESULT

The following snapshots define the results or outputs that we will get after step by step execution of all the modules of the system.

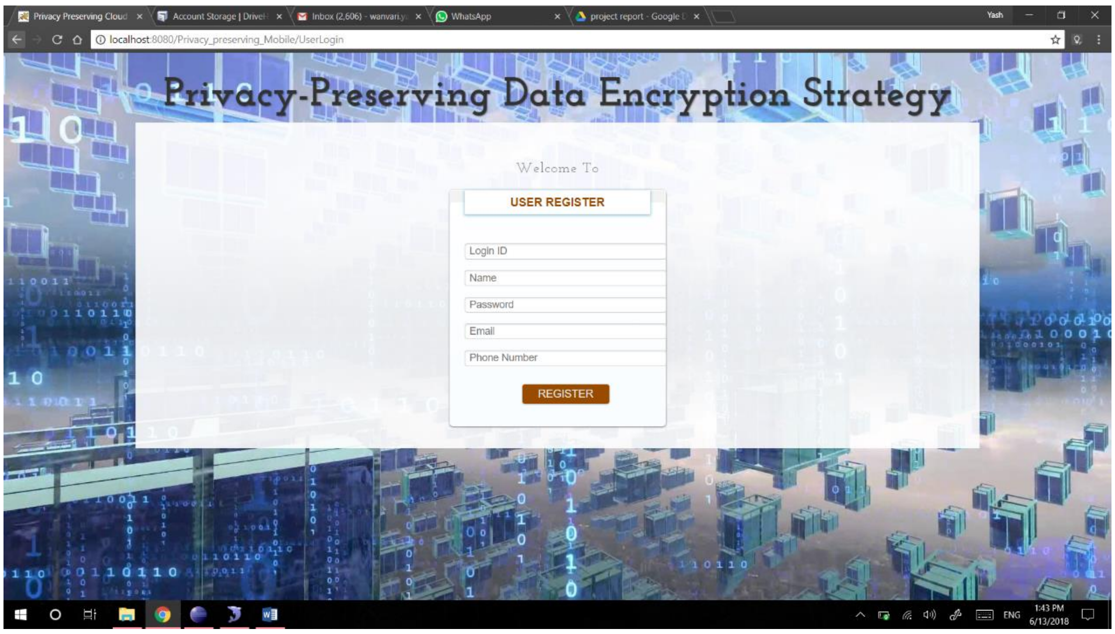


Fig 8. 1 User Registration

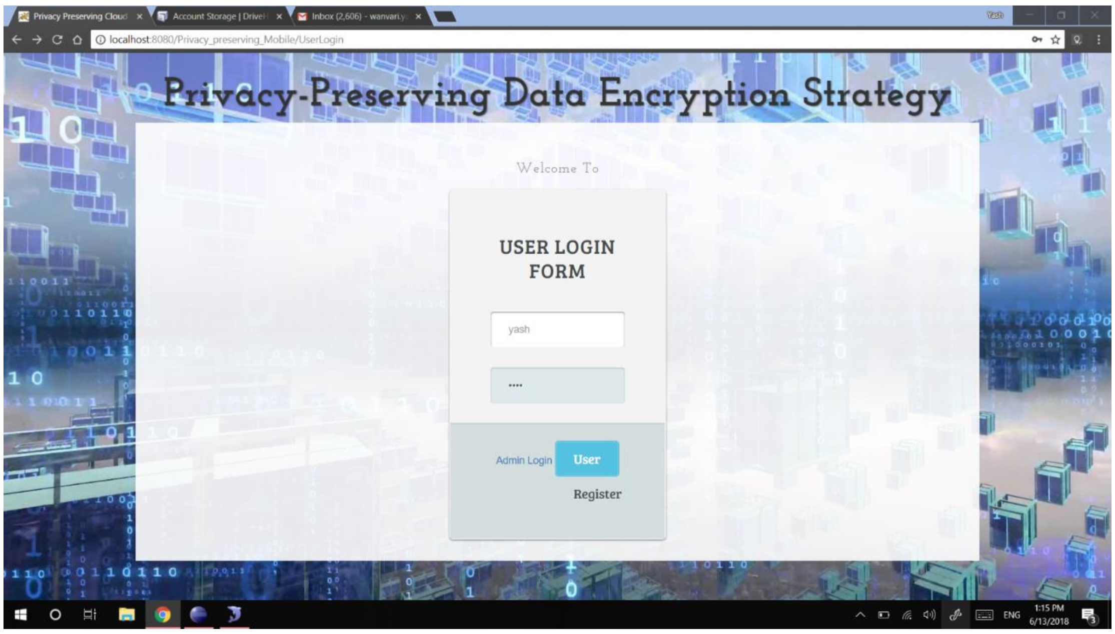
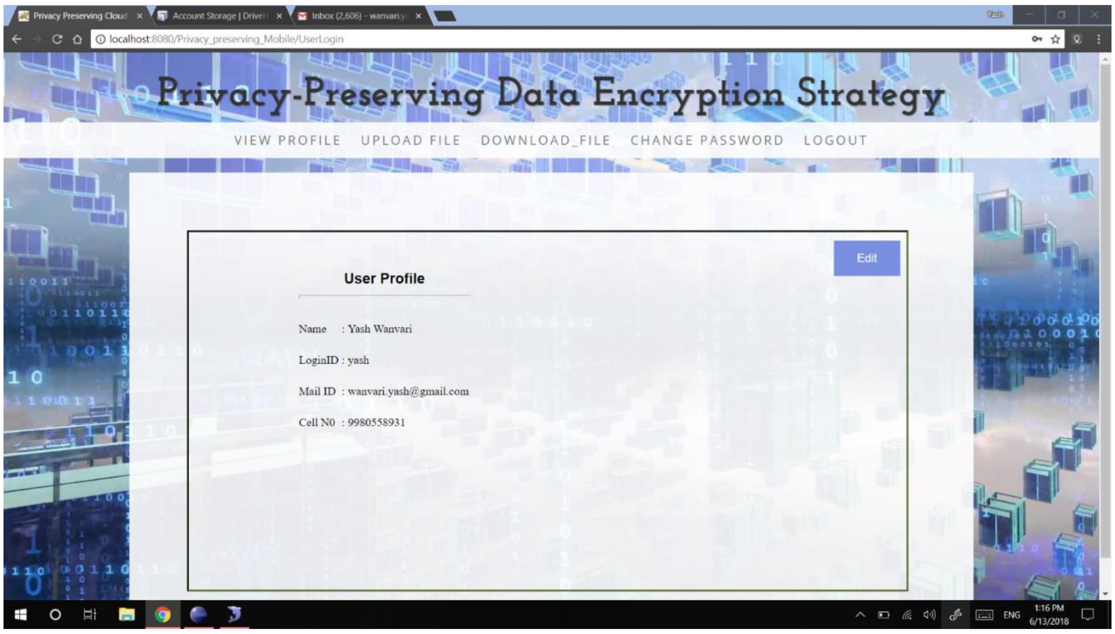

localhost:8080/Privacy_preserving_Mobile/UserLogin
Privacy-Preserving Data Encryption Strategy
Welcome To
USER LOGIN FORM
yash
••••
Admin Login
User
Register

Fig 8. 2: User Login


localhost:8080/Privacy_preserving_Mobile/UserLogin
Privacy-Preserving Data Encryption Strategy
VIEW PROFILE
UPLOAD FILE
DOWNLOAD_FILE
CHANGE PASSWORD
LOGOUT
Edit

User Profile

Name : Yash Wanvari

LoginID : yash

Mail ID : wanvari.yash@gmail.com

Cell N0 : 9980558931

Fig 8. 3: User Profile

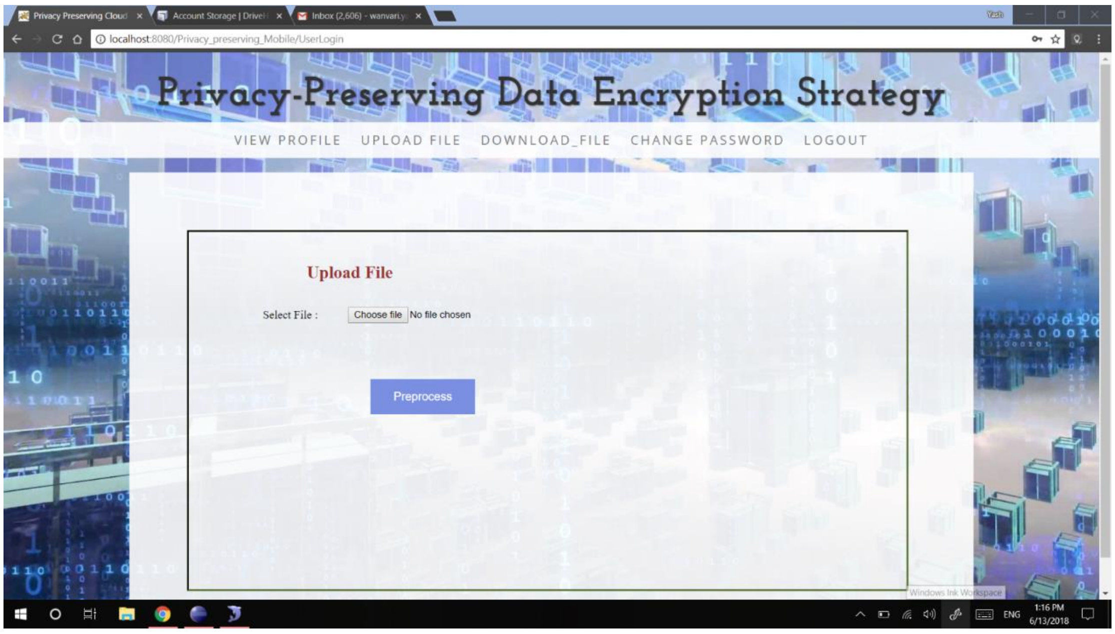
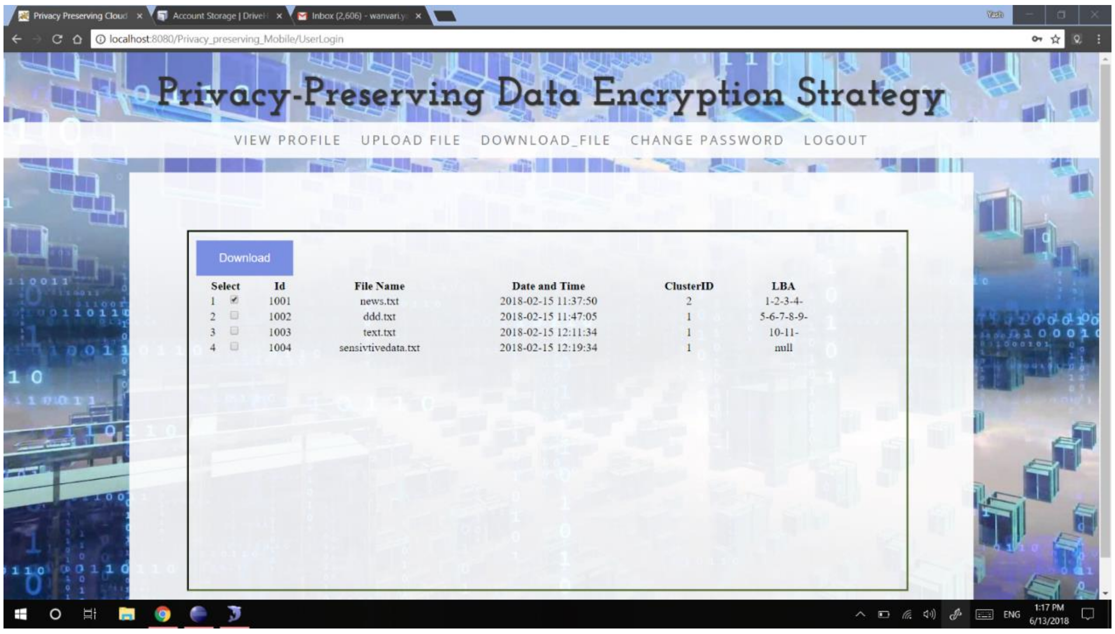


Fig 8. 4: File Upload Options for the User



Select		Id	File Name	Date and Time	ClusterID	LBA
1	✓	1001	news.txt	2018-02-15 11:37:50	2	1-2-3-4-
2	☐	1002	ddd.txt	2018-02-15 11:47:05	1	5-6-7-8-9-
3	☐	1003	text.txt	2018-02-15 12:11:34	1	10-11-
4	☐	1004	sensivtivedata.txt	2018-02-15 12:19:34	1	null

Fig 8. 5: File Download options for the User

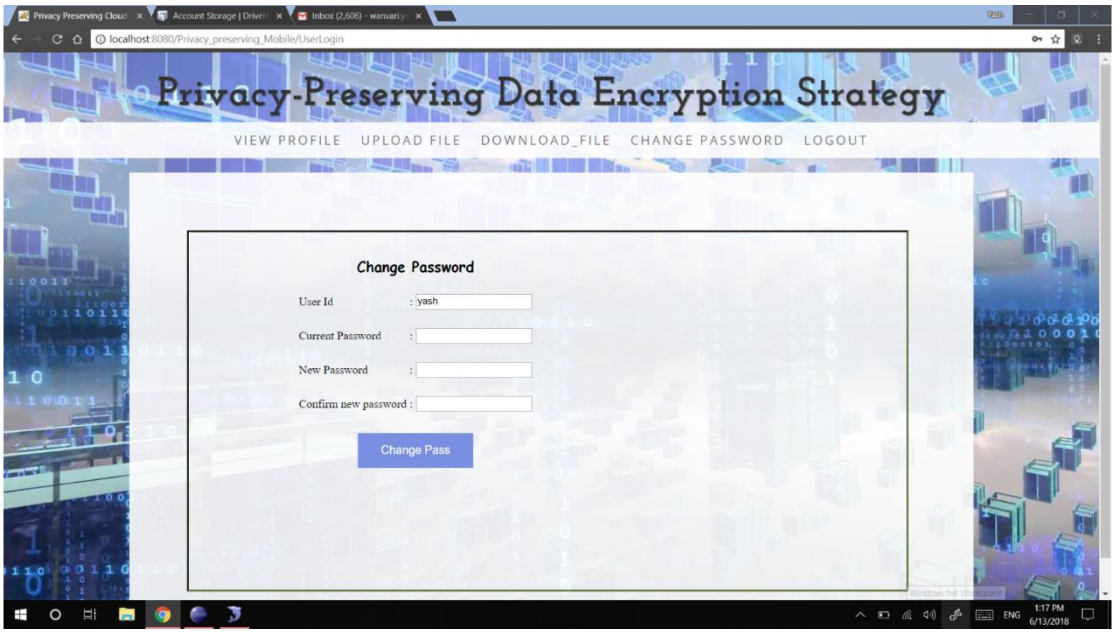


Fig 8. 6: The Change Password option for User

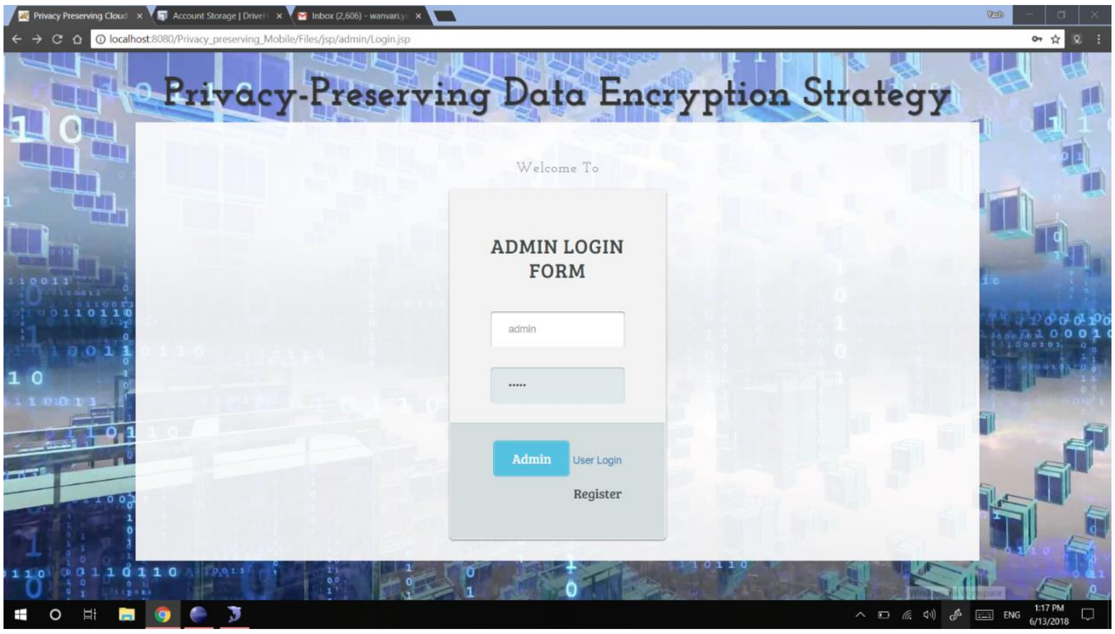
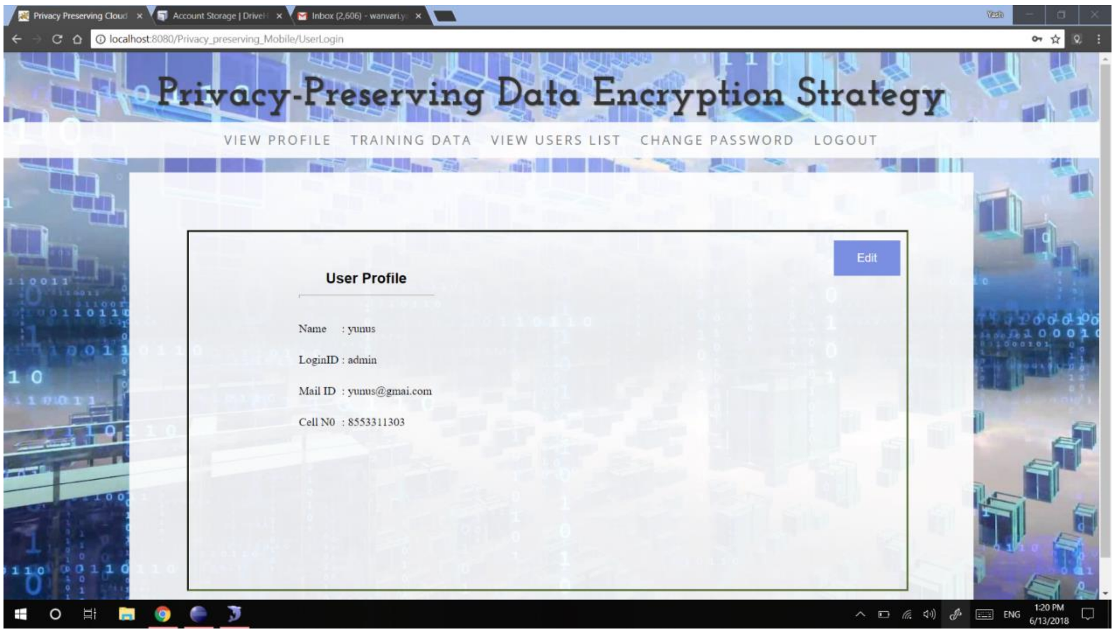

localhost:8080/Privacy_preserving_Mobile/Files/jsp/admin/Login.jsp
Privacy-Preserving Data Encryption Strategy
Welcome To
ADMIN LOGIN FORM
admin
Admin
User Login
Register

Fig 8. 7: Admin Login


localhost:8080/Privacy_preserving_Mobile/UserLogin
Privacy-Preserving Data Encryption Strategy
VIEW PROFILE
TRAINING DATA
VIEW USERS LIST
CHANGE PASSWORD
LOGOUT
Edit

User Profile

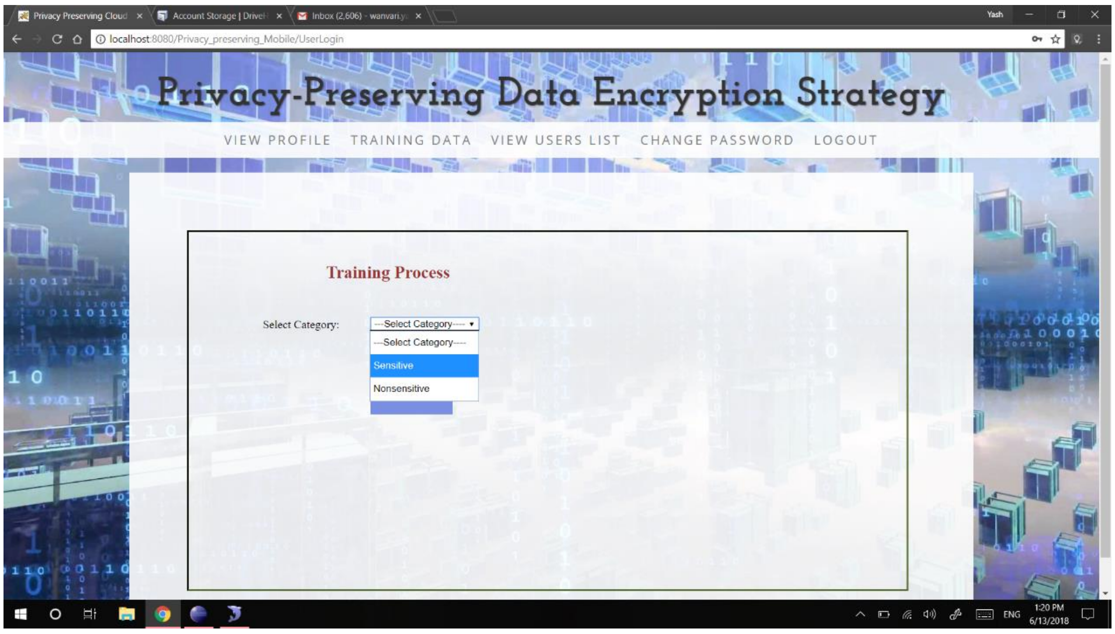
Name : yunus

LoginID : admin

Mail ID : yunus@gmai.com

Cell N0 : 8553311303

Fig 8. 8: Admin's Profile

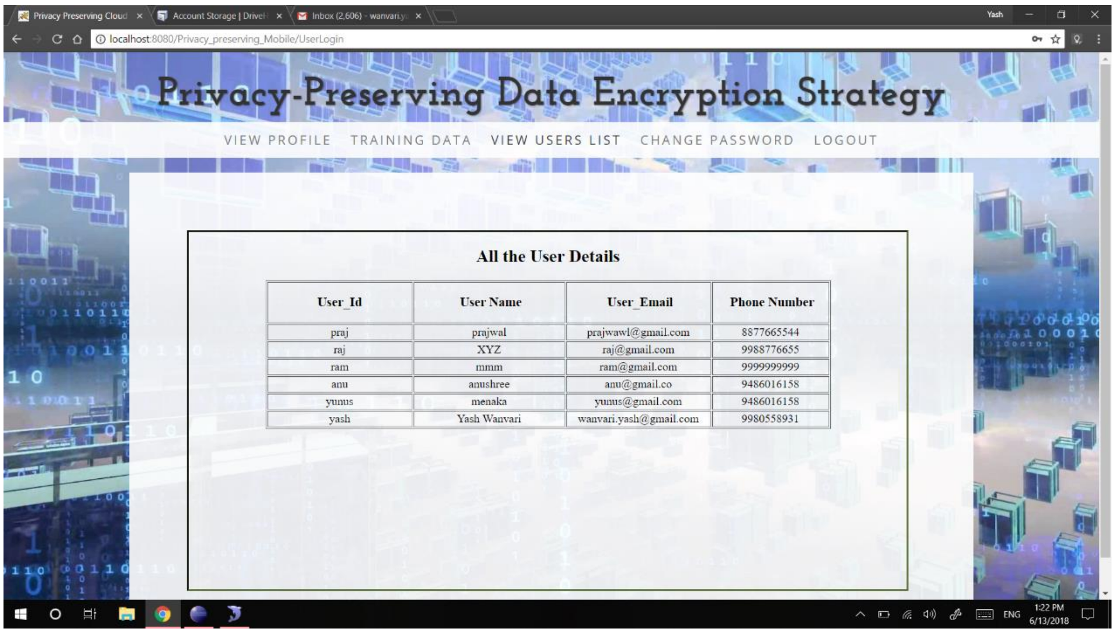

localhost:8080/Privacy_preserving_Mobile/UserLogin
Privacy-Preserving Data Encryption Strategy
VIEW PROFILE
TRAINING DATA
VIEW USERS LIST
CHANGE PASSWORD
LOGOUT

Training Process

Select Category: ----Select Category----

----Select Category----
Sensitive
Nonsensitive

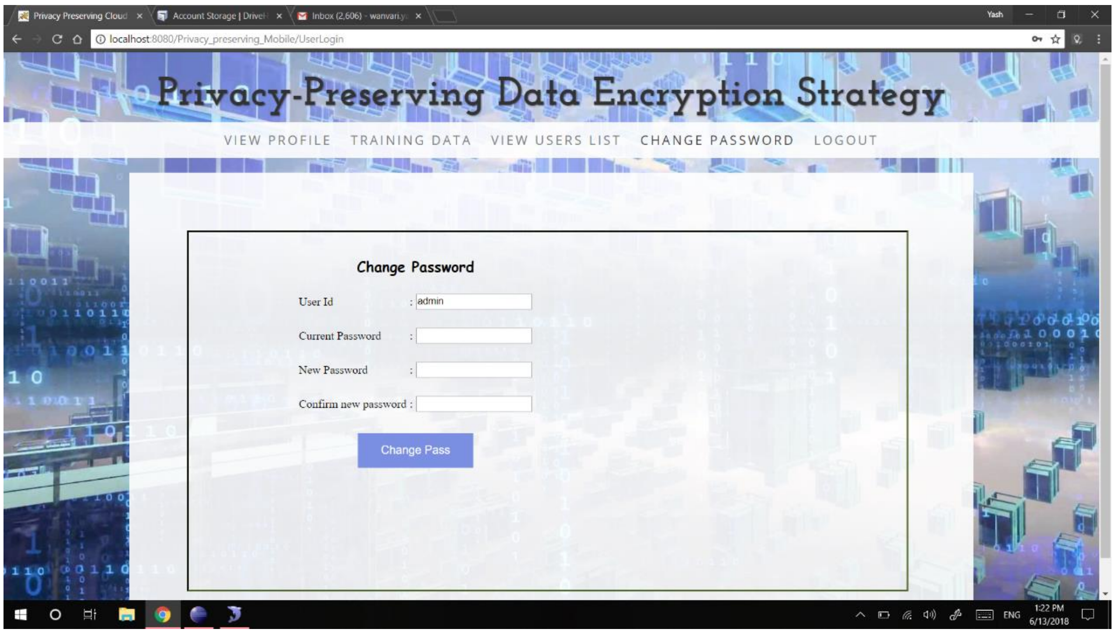
Fig 8. 9: Admin's Data Processing Options



All the User Details

User_Id	User Name	User_Email	Phone Number
praj	prajwal	prajwawl@gmail.com	8877665544
raj	XYZ	raj@gmail.com	9988776655
ram	mmm	ram@gmail.com	9999999999
anu	anushree	anu@gmail.co	9486016158
yunus	menaka	yunus@gmail.com	9486016158
yash	Yash Wanvari	wanvari.yash@gmail.com	9980558931

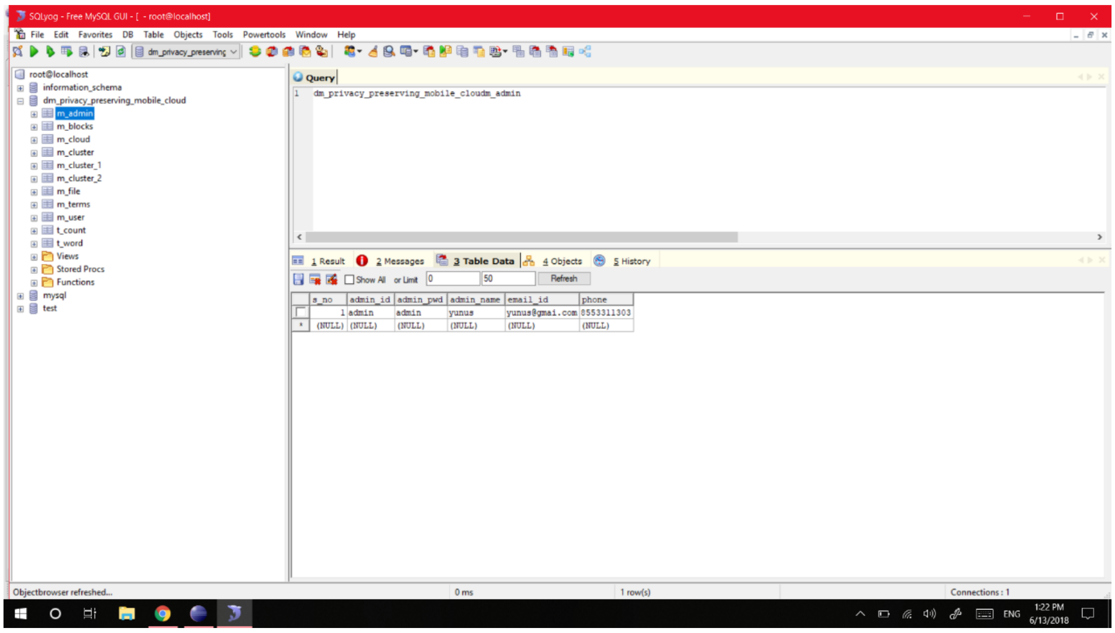
Fig 8. 10: Admin can view User Detail



Change Password

User Id	: admin
Current Password	:
New Password	:
Confirm new password	:

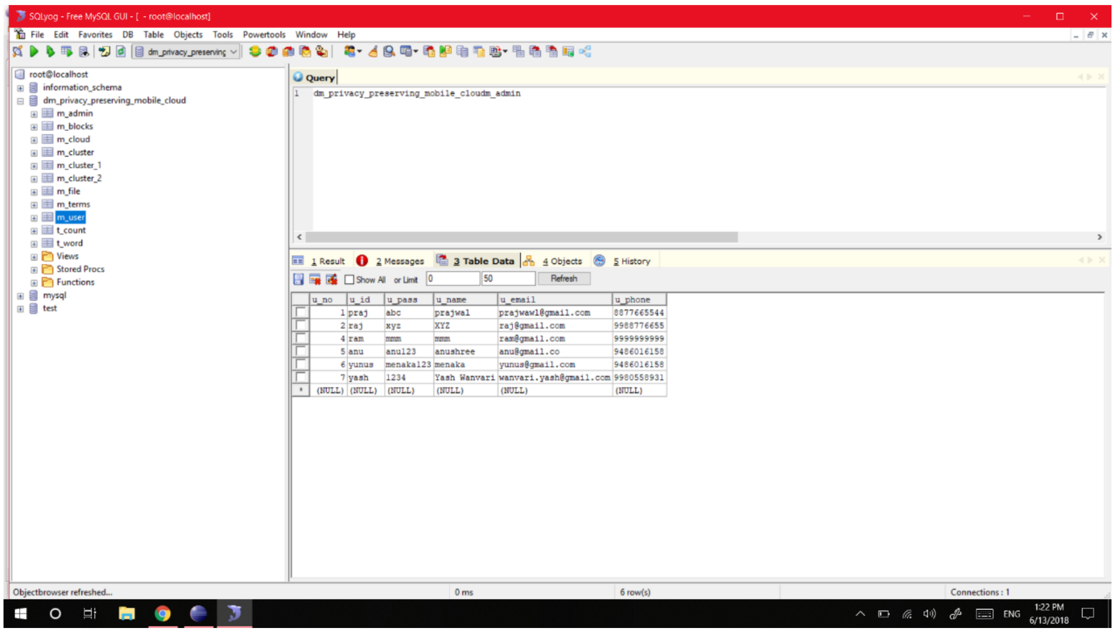
Fig 8. 11: Changing Admin Password



```
dm_privacy_preserving_mobile_cloudm_admin
```

s_no	admin_id	admin_pwd	admin_name	email_id	phone
1	admin	admin	yunus	yunus@gmai.com	8553311303
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

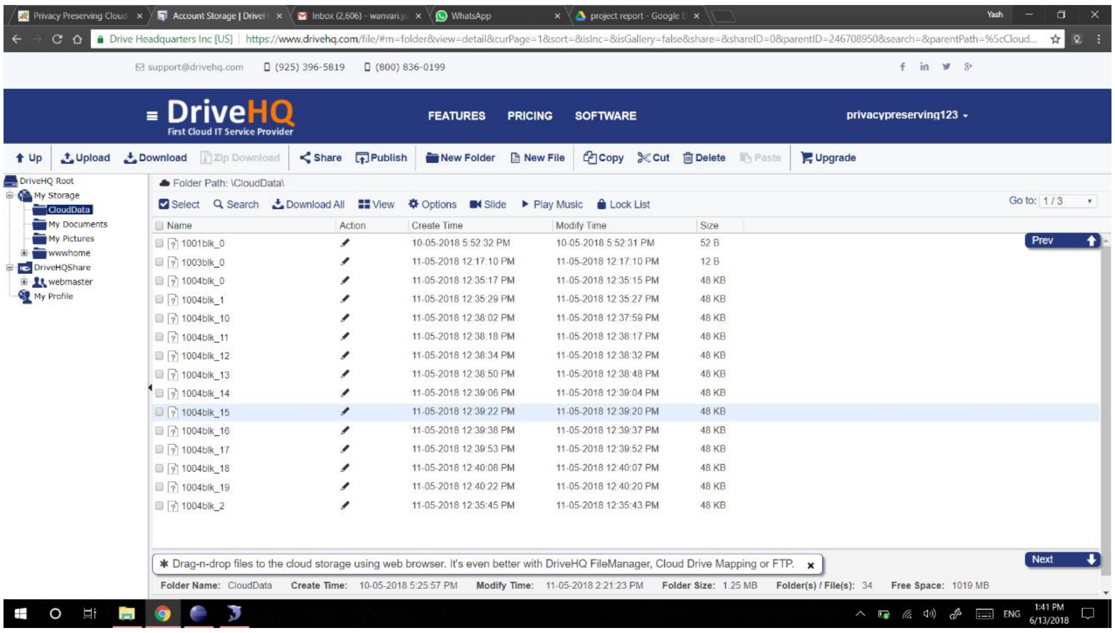
Fig 8. 12: SQL database showing Admin Information



```
dm_privacy_preserving_mobile_cloudm_admin
```

u_no	u_id	u_pass	u_name	u_email	u_phone
1	praj	abc	prajwal	prajwawl@gmail.com	8877665544
2	raj	xyz	XYZ	raj@gmail.com	9988776655
4	ram	mmm	mmm	ram@gmail.com	9999999999
5	anu	anu123	anushree	anu@gmail.co	9486016158
6	yunus	menaka123	menaka	yunus@gmail.com	9486016158
7	yash	1234	Yash Wanvari	wanvari.yash@gmail.com	9980558931
(NULL)	(NULL)	(NULL)	(NULL)	(NULL)	(NULL)

Fig 8. 13: SQL database show User Information



Folder Path: \CloudData\

Name	Action	Create Time	Modify Time	Size
1001blk_0		10-05-2018 5:52:32 PM	10-05-2018 5:52:31 PM	52 B
1003blk_0		11-05-2018 12:17:10 PM	11-05-2018 12:17:10 PM	12 B
1004blk_0		11-05-2018 12:35:17 PM	11-05-2018 12:35:15 PM	48 KB
1004blk_1		11-05-2018 12:35:29 PM	11-05-2018 12:35:27 PM	48 KB
1004blk_10		11-05-2018 12:38:02 PM	11-05-2018 12:37:59 PM	48 KB
1004blk_11		11-05-2018 12:38:18 PM	11-05-2018 12:38:17 PM	48 KB
1004blk_12		11-05-2018 12:38:34 PM	11-05-2018 12:38:32 PM	48 KB
1004blk_13		11-05-2018 12:38:50 PM	11-05-2018 12:38:48 PM	48 KB
1004blk_14		11-05-2018 12:39:06 PM	11-05-2018 12:39:04 PM	48 KB
1004blk_15		11-05-2018 12:39:22 PM	11-05-2018 12:39:20 PM	48 KB
1004blk_16		11-05-2018 12:39:38 PM	11-05-2018 12:39:37 PM	48 KB
1004blk_17		11-05-2018 12:39:53 PM	11-05-2018 12:39:52 PM	48 KB
1004blk_18		11-05-2018 12:40:08 PM	11-05-2018 12:40:07 PM	48 KB
1004blk_19		11-05-2018 12:40:22 PM	11-05-2018 12:40:20 PM	48 KB
1004blk_2		11-05-2018 12:35:45 PM	11-05-2018 12:35:43 PM	48 KB

✱ Drag-n-drop files to the cloud storage using web browser. It's even better with DriveHQ FileManager, Cloud Drive Mapping or FTP.

Folder Name: CloudData **Create Time:** 10-05-2018 5:25:57 PM **Modify Time:** 11-05-2018 2:21:23 PM **Folder Size:** 1.25 MB **Folder(s) / File(s):** 34 **Free Space:** 1019 MB

Fig 8. 14: Files split into blocks uploaded on the DriveHQ Server

Chapter 9

CONCLUSION

This paper focused on the privacy issues of big data and considered the practical implementations in cloud computing. The proposed approach, D2ES, was designed to maximize the efficiency of privacy protections. Main algorithm supporting D2ES model was DED algorithm that was developed to dynamically alternative data packages for encryptions under different timing constraints. The experimental evaluations showed the proposed approach had an adaptive and superior performance.

References

[1] S. Yu, W. Zhou, S. Guo, and M. Guo. A feasible IP traceback framework through dynamic deterministic packet marking. IEEE Transactions on Computers, 65(5):1418–1427, 2016.

[2] S. Yu, G. Gu, A. Barnawi, S. Guo, and I. Stojmenovic. Malware propagation in large-scale networks. IEEE Transactions on Knowledge and Data Engineering, 27(1):170–179, 2015.

[3] S. Liu, Q. Qu, L. Chen, and L. Ni. SMC: A practical schema for privacy-preserved data sharing over distributed data streams. IEEE Transactions on Big Data, 1(2):68–81, 2015.

[4] S. Rho, A. Vasilakos, and W. Chen. Cyber physical systems technologies and applications. Future Generation Computer Systems, 56:436– 437, 2016.

[5] L. Wu, K. Wu, A. Sim, M. Churchill, J. Choi, A. Stathopoulos, C. Chang, and S. Klasky. Towards real-time detection and tracking of spatio-temporal features: Blob-filaments in fusion plasma. IEEE Transactions on Big Data, 2(3), 2016.

[6] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar. Dependable demand response management in the smart grid: A stackelberg game approach. IEEE Transactions on Smart Grid, 4(1):120–132, 2013.

[7] M. Qiu, M. Zhong, J. Li, K. Gai, and Z. Zong. Phase-change memory optimization for green cloud with genetic algorithm. IEEE Transactions on Computers, 64(12):3528–3540, 2015.

[8] H. Liu, H. Ning, Y. Zhang, Q. Xiong, and L. Yang. Role-dependent privacy preservation for secure V2G networks in the smart grid. IEEE Transactions on Information Forensics and Security, 9(2):208–220, 2014.

[9] F. Tao, Y. Cheng, D. Xu, L. Zhang, and B. Li. CCIoT-CMfg: cloud computing and internet of things-based cloud manufacturing service system. IEEE Transactions on Industrial Informatics, 10(2):1435– 1442, 2014.

[10] G. Wu, H. Zhang, M. Qiu, Z. Ming, J. Li, and X. Qin. A decentralized approach for mining event correlations in distributed system monitoring. Journal of parallel and Distributed Computing, 73(3):330–340, 2013.

[11] S. Yu, W. Zhou, R. Doss, and W. Jia. Traceback of DDoS attacks using entropy variations. IEEE Transactions on Parallel and Distributed Systems, 22(3):412–425, 2011.

[12] Y. Li, W. Dai, Z. Ming, and M. Qiu. Privacy protection for preventing data over-collection in smart city. IEEE Transactions on Computers, 65:1339–1350, 2015.

[13] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang. Discriminating DDoS attacks from flash crowds using flow correlation coefficient. IEEE Transactions on Parallel and Distributed Systems, 23(6):1073– 1080, 2012.

[14] Y. Yu, M. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min. Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage. IEEE Transactions on Information Forensics and Security, PP(99):1, 2016.

[15] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet. A privacy-preserving framework for large-scale content-based information retrieval. IEEE Transactions on Information Forensics and Security, 10(1):152–167, 2015.

[16] K. Gai, M. Qiu, H. Zhao, and J. Xiong. Privacy-aware adaptive data encryption strategy of big data in cloud computing. In The 2nd IEEE International Conference of Scalable and Smart Cloud (SSC 2016), pages 273–278, Beijing, China, 2016. IEEE.

[17] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang. SCLPV: Secure certificateless public verification for cloud-based cyber-physicalsocial systems against malicious auditors. IEEE Transactions on Computational Social Systems, 2(4):159–170, 2015.

[18] C. Wang, S. Chow, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for secure cloud storage. IEEE Transactions on Computers, 62(2):362–375, 2013.

[19] K. Gai, L. Qiu, M. Chen, H. Zhao, and M. Qiu. SA-EAST: securityaware efficient data transmission for ITS in mobile heterogeneous cloud computing. ACM Transactions on Embedded Computing Systems, 16(2):60, 2017.

[20] C. Lai, M. Chen, M. Qiu, A. Vasilakos, and J. Park. A RF4CEbased remote controller with interactive graphical user interface applied to home automation system. ACM Transactions on Embedded Computing Systems, 12(2):30, 2013.

[21] S. Backhaus, R. Bent, J. Bono, R. Lee, B. Tracey, D. Wolpert, D. Xie, and Y. Yildiz. Cyber-physical security: A game theory model of humans interacting over control systems. IEEE Transactions on Smart Grid, 4(4):2320–2327, 2013.

[22] K. Gai, M. Qiu, H. Zhao, and W. Dai. Privacy-preserving adaptive multi-channel communications under timing constraints. In The IEEE International Conference on Smart Cloud 2016, page 1, New York, USA, 2016. IEEE.

[23] L. Tang, X. Yu, Q. Gu, J. Han, G. Jiang, A. Leung, and T. Porta. A framework of mining trajectories from untrustworthy data in cyberphysical system. ACM Transactions on Knowledge Discovery from Data, 9(3):16, 2015.

[24] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich. VC3: Trustworthy data analytics in the cloud using SGX. In IEEE Symposium on Security and Privacy, pages 38–54, San Jose, CA, USA, 2015. IEEE.

[25] M. Maffei, G. Malavolta, M. Reinert, and D. Schroder. Privacy and access control for outsourced personal records. In IEEE Symposium on Security and Privacy, pages 341–358, San Jose, CA, USA, 2015. IEEE.

[26] Y. Li, K. Gai, Z. Ming, H. Zhao, and M. Qiu. Intercrossed access control for secure financial services on multimedia big data in cloud systems. ACM Transactions on Multimedia Computing Communications and Applications, 12(4s):67, 2016.

[27] C. Mulliner, W. Robertson, and E. Kirda. Hidden GEMs: Automated discovery of access control vulnerabilities in graphical user interfaces. In IEEE Symposium on Security and Privacy, pages 149–162, San Jose, CA, USA, 2014. IEEE.

[28] S. Sen, S. Guha, A. Datta, S. Rajamani, J. Tsai, and J. Wing. Bootstrapping privacy compliance in big data systems. In IEEE Symposium on Security and Privacy, pages 327–342, San Jose, CA, USA, 2014. IEEE.

[29] J. Vilk, D. Molnar, B. Livshits, E. Ofek, C. Rossbach, A. Moshchuk, H. Wang, and R. Gal. SurroundWeb: Mitigating privacy concerns in a 3D web browser. In IEEE

Symposium on Security and Privacy, pages 431–446, San Jose, CA, USA, 2015. IEEE.

[30] L. Zhu, Z. Hu, J. Heidemann, D. Wessels, A. Mankin, and N. Somaiya. Connection-oriented DNS to improve privacy and security. In IEEE Symposium on Security and Privacy, pages 171–186, San Jose, CA, USA, 2015. IEEE.

[31] M. Baran, P. Carpenter, L. Borbye, D. Lubkeman, M. Ligett, and D. Covington. A new professional science master program for electric power systems engineering. IEEE Transactions on Power Systems, 29(4):1903–1910, 2014.

[32] K. Gai and A. Steenkamp. A feasibility study of Platform-as-aService using cloud computing for a global service organization. Journal of Information System Applied Research, 7:28–42, 2014.

[33] K. Li, W. Zhang, C. Yang, and N. Yu. Security analysis on one-tomany order preserving encryption-based cloud data search. IEEE Transactions on Information Forensics and Security, 10(9):1918– 1926, 2015.

[34] C. Cicconetti, L. Lenzini, A. Lodi, S. Martello, E. Mingozzi, and M. Monaci. Efficient two-dimensional data allocation in IEEE 802.16 OFDMA. IEEE/ACM Transactions on Networking, 22(5):1645–1658, 2014.

[35] M. Vidyasagar. A metric between probability distributions on finite sets of different cardinalities and applications to order reduction. IEEE Transactions on Automatic Control, 57(10):2464–2477, 2012.

[36] M. Qiu, Z. Chen, Z. Ming, X. Qin, and J. Niu. Energy-aware data allocation with hybrid memory for mobile cloud systems. IEEE Systems Journal, PP:1–10, 2014.

[37] P. Lacroute. Real-time volume rendering on shared memory multiprocessors using the shear-warp factorization. In Proceedings of the IEEE symposium on Parallel rendering, pages 15–22, Atlanta, GA, USA, 1995. ACM.