HIDDEN CIPHERTEXT POLICY ATTRIBUTE-BASED ENCRYPTION WITH FAST DECRYPTION FOR PERSONAL HEALTH RECORD SYSTEM BY USING CLOUD COMPUTING

A MAJOR PROJECT REPORT

Submitted by

V.SNEHA (19841A05K8)

M.JAGADEESH REDDY (19841A05N3)

SUMIT KUMAR YADAV (19841A05L8)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

AURORA'S TECHNOLOGICAL AND RESEARCH INSTITUTE

(Approved by AICTE and Affiliated to JNTUH) (Accredited by NAAC with 'A' Grade)

Parvathapur, Uppal, Medipally (M), Medchal(D). Hyderabad-500098

JUNE 2023

AURORA'S TECHNOLOGICAL & RESEARCH INSTITUTE

(Approved by AICTE and Affiliated to JNTUH) (Accredited by NAAC with 'A' Grade)

Parvathapur, Uppal, Medipally (M), Medchal(D). Hyderabad-500039

DECLARATION

We hereby declare that the work described in this project, entitled '**HIDDEN CIPHERTEXT POLICY ATTRIBUTE-BASED ENCRYPTION WITH FAST DECRYPTION FOR PERSONAL HEALTH RECORD SYSTEM BY USING CLOUD COMPUTING**' which is being submitted by us in partial fulfillment for award of Bachelor of Technology in Computer Science and Engineering to **AURORA'S TECHNOLOGICAL AND RESEARCH INSTITUTE** is the result of investigation carried by us under the guidance of **Mr. Ch. Krishna Rao**, **Assistant professor, CSE**.

The work is original and has not been submitted for any degree of this or any other university.

Place: Hyderabad

Date:

V. Sneha (19841A05k8)

M. Jagadeesh Reddy (19841A05N3)

Sumit kumar Yadav(19841A05L8)

AURORA'S TECHNOLOGICAL & RESEARCH INSTITUTE

(Approved by AICTE and Affiliated to JNTUH) (Accredited by NAAC with 'A'Grade)

Parvathapur, Uppal, Medipally (M), Medchal(D). Hyderabad-500039

BONAFIDE CERTIFICATE

Certified that this project "**HIDDEN CIPHERTEXT POLICY ATTRIBUTE-BASED ENCRYPTION WITH FAST DECRYPTION FOR PERSONAL HEALTH RECORD SYSTEM BY USING CLOUD COMPUTING**" is the bonafide work of **"V. SNEHA (19841A05K8), M. JAGADEESH REDDY (19841A05N3), J. SUMIT KUMAR YADAV (19841A05L8)**" who carried out the project work under our supervision.

GUIDE
Mr. Ch. Krishna Rao
Associate Professor
Dept. of CSE/IT

HEAD OF THE DEPARTMENT
Ms. Durga Pavani
Dept. of CSE

MINI PROJECT COORDINATOR
Mr. Ch. Krishna Rao
Assistant Professor
Dept. of CSE/IT

DIRECTOR
Mr. Srikanth Jatla

EXTERNAL EXAMINER

ACKNOWLEDEGMENT

This work has been done during the project period and it was a very good opportunity to put theoretical knowledge into planned exercise with an aim to solve a real time problem and also to develop confidence to face various practical situations.

We would also like to express our gratitude to **Mr. Srikanth Jatla, Director,** Aurora's Technological and Research Institute for providing us with a congenial atmosphere and encouragement.

We express our sincere thanks to Head of the Department **Ms. Durga Pavani** for giving us the support and her kind attention and valuable guidance to us throughout this course.

We express our sincere thanks to Project Coordinator **Mr. Ch. Krishna Rao** for helping us to complete our project work by giving valuable suggestions.

We convey thanks to our project guide **Mr. Ch. Krishna Rao**, Department of Computer Science and Engineering, for providing encouragement, constant support and guidance which was of great help to complete this project successfully.

V.SNEHA (19841A05K8)

M.JAGADEESH REDDY (19841A05N3)

SUMIT KUMAR YADAV (19841A05L8)

`ABSTRACT

Since cloud computing has been playing an increasingly important role in real life, the privacy protection in many fields has been paid more and more attention, especially, in the field of Personal Health Record (PHR). The traditional ciphertext-policy attribute-based encryption (CP-ABE) provides the fine-grained access control policy for encrypted PHR data, but the access policy is also sent along with ciphertext explicitly. However, the access policy will reveal the users' privacy because it contains too much sensitive information of the legitimate data users. Hence it is important to protect users' privacy by hiding access policies. In most of the previous schemes, although the access policy is hidden, they facet wo practical problems: (1) these schemes do not support large attribute universe, so their practicality in PHR is greatly limited, and (2) the cost of decryption is especially high since the access policy is embedded in ciphertext. To address these problems, we construct a CP-ABE scheme with efficient decryption, where both the size of public parameters and the cost of decryption are constant. Moreover, we also show the proposed scheme achieves full security in the standard model under static assumptions by using the dual system encryption method.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1. PURPOSE

cloud computing provides a fast and efficient way to share data resources, and a mountain number of people access data through the network. For example, in the personal system health record system, a patient does not have to carry various paper versions of the test forms to make a diagnosis according to the traditional way, but he/she can store, retrieve and share the health record only by uploading his own personal health record to the PHR system. A patient has the full control to his/her own PHR document and authorizes who can access these health data, such as friends, family or healthcare providers. In order to achieve accurate access control of PHR, data owners urgently need a kind of encryption scheme that can realize fine grained access control [22], [28], [30], [32].

Hidden ciphertext policy attribute-based encryption scheme provides a good way to solve the problem, where it achieves privacy protection by hiding access control policy. However, In the previous mechanisms [2], [3], [7] , the access control policy is often sent along with ciphertext explicitly, which makes it easy reveal the users' privacy, since some attributes in access structure carry crucial identity information of the legitimate users. In PHR, an access policy defined by a patient may contain some sensitive attributes such as cardiologist, central hospital and so on [8], [31] . Therefore, for an unauthorized user, even if he cannot decrypt successfully, he can also infer from the access policy in cleartext form which the encryptor suffers from some disease. The first hidden ciphertext-policy attribute-based encryption (HCP-ABE) was introduced in [16], where the access structure was embedded in the ciphertext and not sent directly. Subsequently, some other hidden CP-ABE schemes were also successively proposed in [17]– [19]. However, access structures in these schemes only support AND gates or AND gates on positive, negative and wildcard. These lead to two drawbacks. First, the size of public parameters increases linearly with the number of attributes, and secondly, the cost of the decryption is greatly increased. Due to the above drawbacks, some low-overhead schemes are introduced in [13]– [14] and the common method adopted by these schemes is to introduce a decryption test by adding some redundant components to a ciphertext before the decryption stage. Although the above schemes improve the efficiency of decryption, the length of ciphertext is also significantly increased and this will become a bottleneck restricting higher performance.

1.2 SCOPE

The first CP-ABE scheme was introduced in [7], where ciphertexts were associated with access structure defined by data owners and the key are associated with sets of attributes about users. Subsequently, there are a lot of CP-ABE schemes were also successively proposed in [15], [17], [18], and [21], but these schemes only support AND gates. To realize the access structure more expressive, Waters proposed an access structure based a linear secret sharing scheme (LSSS), and it is also a provably secure scheme under the standard model [3]. In order to further protect users' privacy, the first the CP-ABE scheme with hidden access structure was proposed by Nishide et al. [16]. In their work, access control policy isn't sent along with ciphertext explicitly, in other words, no unauthorized user can obtain useful information about the access structure. Some other schemes with the same performance have been proposed by other researchers, which are called Anonymous Attribute-Based Encryption [22], [29]. In these schemes, only sets of the user satisfying the access policy was embedded in the ciphertext, then the user can successfully decrypt the ciphertext. Later, authors in introduced another highly effective anonymous CP-ABE scheme, and its security proof was given under the Decisional Modified Bilinear Diffie-Hellman assumption (MBDH) [20]. However, their work only gives a general analysis and lacks detailed security proof. Some other works were proposed in [9], [14], and [27] to make further improvements on anonymous CP-ABE scheme. Unfortunately, all of them have to face high-overhead of decryption, which may make them lose their practicability.

CHAPTER 2

LITERATURE SURVEY

2.1 A Ciphertext-Policy Attribute-based Encryption Scheme with Optimized Ciphertext Size and Fast Decryption.

Author: M. Qutaibah, S. Abdullatif, and C.T. Viet.

We address the problem of ciphertext-policy attribute-based encryption with fine access control, a cryptographic primitive which has many concrete application scenarios such as Pay-Tv, e-Health, Cloud Storage and so on. In this context we improve on previous LSSS based techniques by building on previous work of Hohenberger and Waters at PKC'13 and proposing a construction that achieves ciphertext size linear in the minimum between the size of the Boolean access formula and the number of its clauses. Our construction also supports fast decryption. We also propose two interesting extensions: the first one aims at reducing storage and computation at the user side and is useful in the context of lightweight devices or devices using a cloud operator. The second proposes the use of multiple authorities to mitigate key escrow by the authority.

2.2 Security and privacy in smart health: Efficient policy-hiding attribute-based access control.

Author: Y. Zhang, D. Zheng, and R.H. Deng.

With the rapid development of the Internet of Things (IoT) and cloud computing technologies, smart health (shealth) is expected to significantly improve the quality of health care. However, data security and user privacy concerns in shealth have not been adequately addressed. As a well-received solution to realize fine-grained access control, ciphertext-policy attribute-based encryption (CP-ABE) has the potential to ensure data security in s-health. Nevertheless, direct adoption of the traditional CP-ABE in s-health suffers two flaws. For one thing, access policies are in cleartext form and reveal sensitive health related information in the encrypted s-health records (SHRs). For another, it usually supports small attribute universe, which places an undesirable limitation on practical deployments of CPABE because the size of its public parameters grows linearly with the size of the universe. To address these problems, we introduce PASH, a privacy-aware s-health access control system, in which the key

ingredient is a large universe CP-ABE with access policies partially hidden. In PASH, attribute values of access policies are hidden in encrypted SHRs and only attribute names are revealed. In fact, attribute values carry much more sensitive information than generic attribute names. Particularly, PASH realizes an efficient SHR decryption test which needs a small number of bilinear pairings. The attribute universe can be exponentially large and the size of public parameters is small and constant. Our security analysis indicates that PASH is fully secure in the standard model. Performance comparisons and experimental results show that PASH is more efficient and expressive than previous schemes.

2.3 Expressive CP-ABE with partially hidden access structures.

Author: J. Lai, R.H. Deng, and Y. Li.

In a traditional ciphertext-policy attribute-based encryption (CP-ABE) scheme, an access structure, also referred to as ciphertext-policy, is sent along with a ciphertext explicitly, and anyone who obtains a ciphertext can know the access structure associated with the ciphertext. In certain applications, access structures contain sensitive information and must be protected from everyone except the users whose private key attributes satisfy the access structures. In this paper, we first propose a new model for CP-ABE with partially hidden access structures. In our model, each attribute consists of two parts: an attribute name and its value; if the private key attributes of a user do not satisfy the access structure associated with a ciphertext, the specific attribute values of the access structure are hidden, while other information about the access structure is public. Based on the CP-ABE scheme proposed by Lewko et al. [14] recently, we then present an efficient construction of CPABE with partially hidden access structures. Compared to previous works in this field, our construction is more flexible and expressive and is proven fully secure in the standard model.

2.4 Ciphertext-Policy Attribute-Based Encryption with Hidden Access Policy and Testing

Author: Yinghui Zhang, Dong Zheng, Robert H. Deng, Published on 2016.

In CP-ABE scheme, the user is able to decrypt the ciphertext only if the attributes defined in the secret key satisfy the access policy specified in the ciphertext. The access policy should be sent to the user along with the corresponding ciphertext.

However, the access policy may contain some sensitive information. Some CP-ABE schemes with hidden access policy were presented to protect the privacy for users. However, in the present CP-ABE schemes with hidden access policy, the users need to do excessive calculation for decryption whether their attributes match the access policy specified in the ciphertext or not, which makes the users do useless computation if the attributes don't match the hidden access policy.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Hidden ciphertext policy attribute-based encryption scheme provides a good way to solve the problem, where it achieves privacy protection by hiding access control policy. However, In the previous mechanisms, the access control policy is often sent along with ciphertext explicitly, which makes it easy reveal the users' privacy, since some attributes in access structure carry crucial identity information of the legitimate users.

3.1.1 DISADVANTAGES OF EXISTING SYSTEM:

- There is a chance to reveal the user's data.
- The size of public parameters increases linearly with the number of attributes.
- The cost of the decryption is greatly increased.

3.2 PROPOSED SYSTEM

We construct a CP-ABE scheme with efficient decryption, where both the size of public parameters and the cost of decryption are constant. Moreover, we also show the proposed scheme achieves full security in the standard model under static assumptions by using the dual system encryption method.

3.2.1 ADVANTAGES OF PROPOSED SYSTEM:

- Sensitive attribute values can be hidden.
- It can protect users' privacy well in PHR.

CHAPTER 4
SOFTWARE REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

4.2 NON-FUNCTIONAL REQUIREMENTS

These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.

They basically deal with issues like:

Portability

Security

Maintainability

Reliability

Scalability

Performance

Reusability

Flexibility

CHAPTER 5

REQUIREMENT SPECIFICATION

5.1 HARDWARE REQUIREMENTS

Speed	-	1.1 Ghz
RAM	-	256 MB (min)
Hard Disk	-	20 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

5.2 SOFTWARE REQIREMENTS

Operating System	:	Windows 10
Application Server	:	Tomcat5.0/6.X
Front End	:	HTML, Css ,Jsp
Scripts	:	JavaScript.
Server side Script	:	Java Server Pages.
Database	:	Mysql
Database Connectivity	:	JDBC.

CHAPTER 6

SOFTWARE DESIGN

6.1 ARCHITECTURAL DESIGN

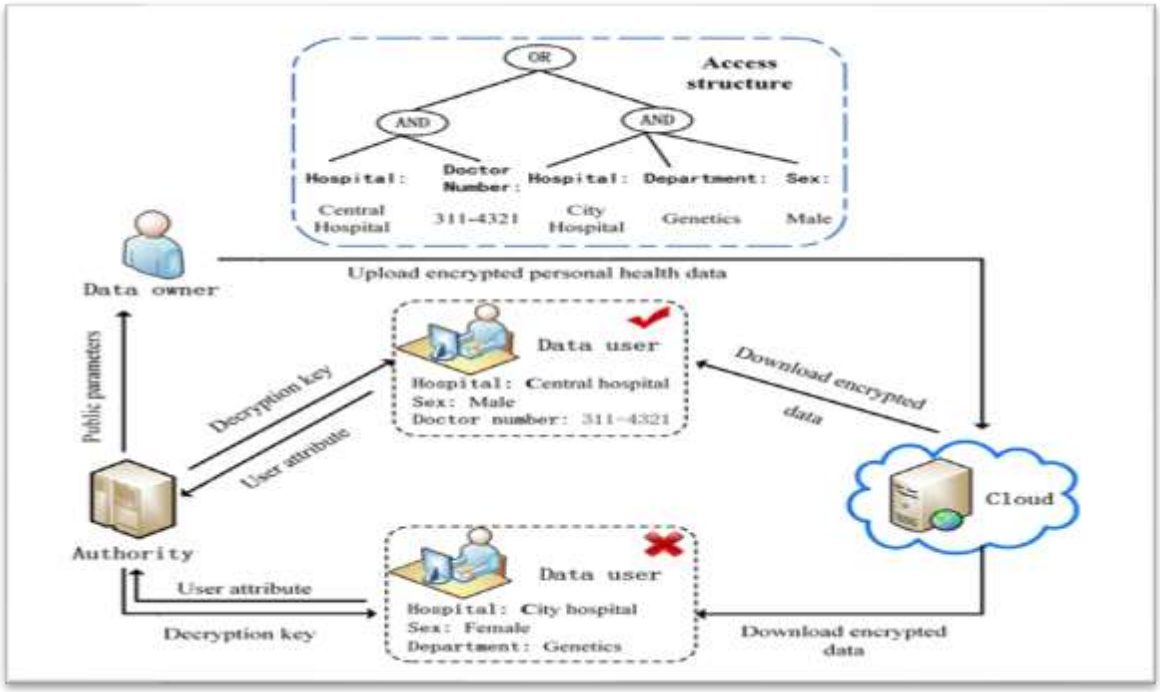


Figure 6.1: System Architecture

6.2 MODULES

1. DATA USER
2. DOCTOR.
3. DATAOWNER
4. AUTHORITY
5. CLOUD SERVER

6.2.1 MODULES DESCRIPTION

- **DATAUSER**

This is the user module, here, the user has to register with the application and the user should be authorized by the cloud server then only the user can access the user home page.

After he logged in his home page the user can perform some actions like view profile, View patient details, view clinical reports.

- **DOCTOR:**

Here doctor also a module, the doctor can able to read the patient details before the doctor should get the access controls from the cloud server. Then the doctor can give the clinical report for the patient.

- **DATAOWNER**

This is the owner module, here, the owner has to register with the application and the owner should be authorized by the cloud server then only the owner can access the owner home page.After he logged in his home page the owner can perform some actions like view profile, add patient details before add the details the owner should get the encrypt key from the authority, view patient details.

- **AUTHORITY**

The authority is one of the modules, authority no need to login with the application, he can directly login with the application and he performs some actions such as generate secret key, and generate encrypt key.

- **CLOUD SERVER**

The cloud server is also a module and the cloud server can access his home page directly without register, the cloud server can perform some actions like: View clinical report, view patient details. authorize users, authorize owner.

6.3 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

6.3.1 USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and 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. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

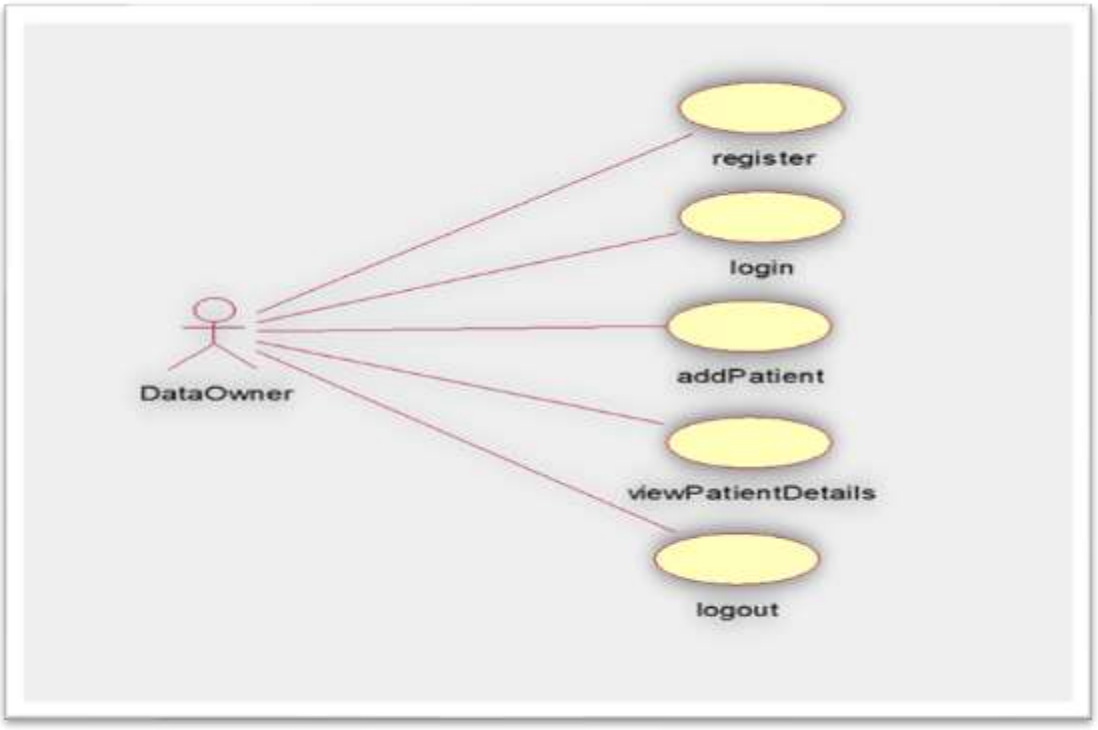


Figure 6.2: use case diagram for data owner

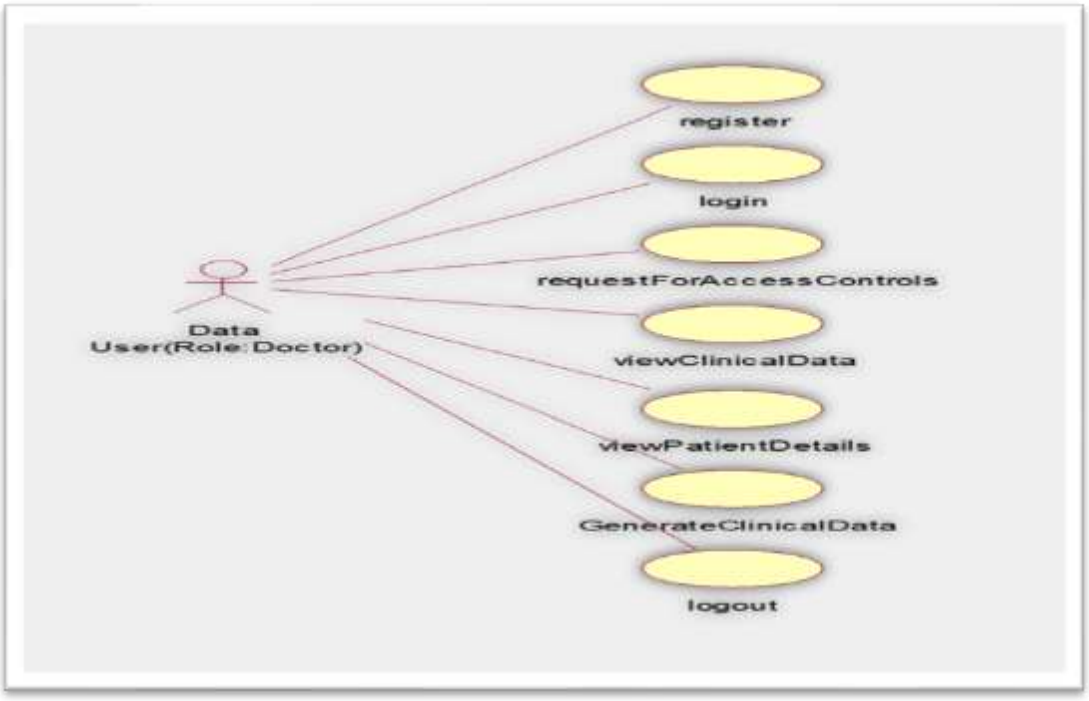


Figure 6.3: use case diagram for data user(doctor)

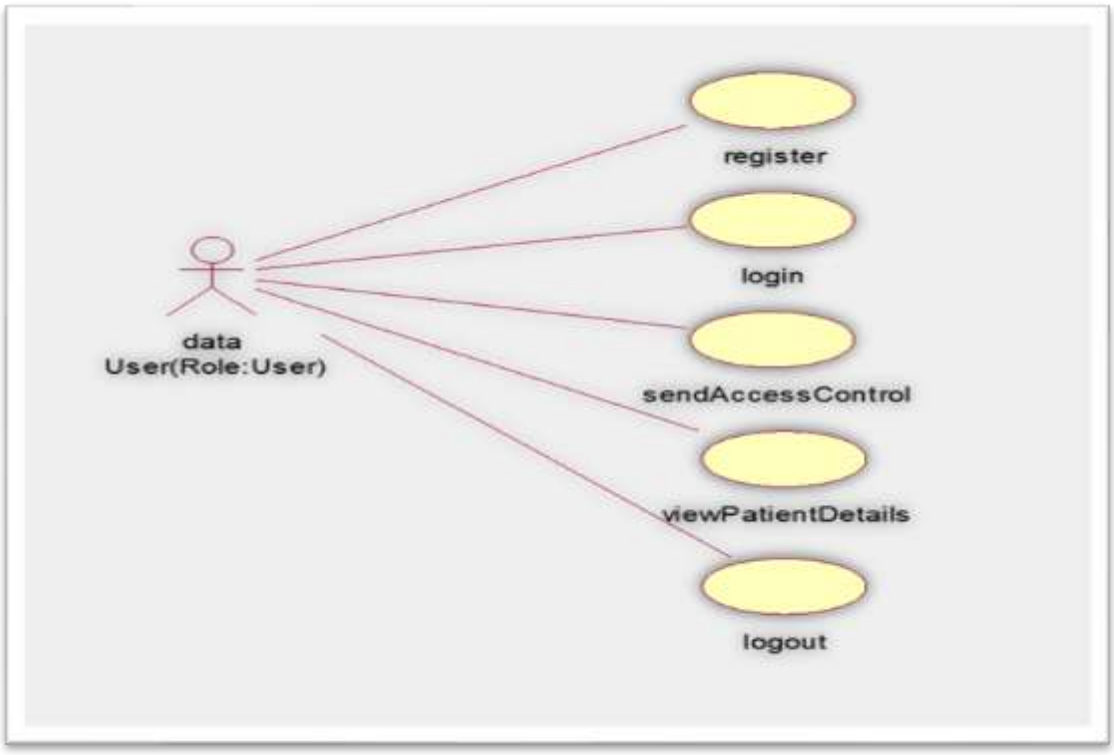


Figure 6.4: use case diagram for data user(user)

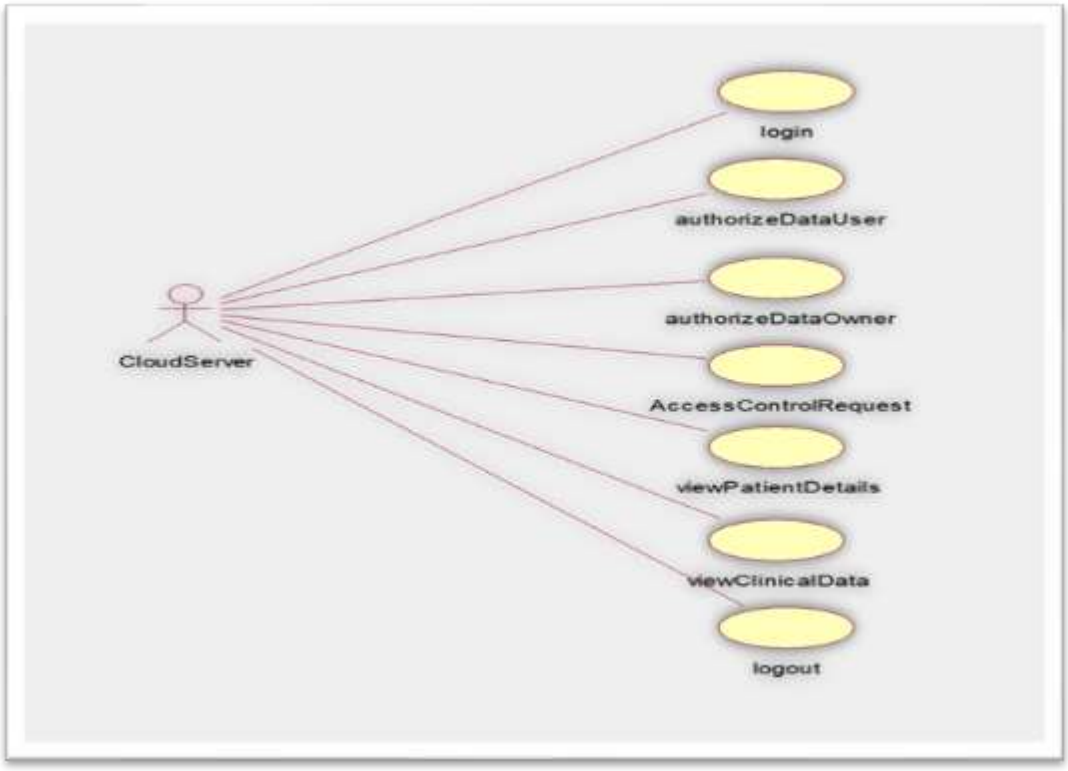


Figure 6.5: use case diagram for cloud server

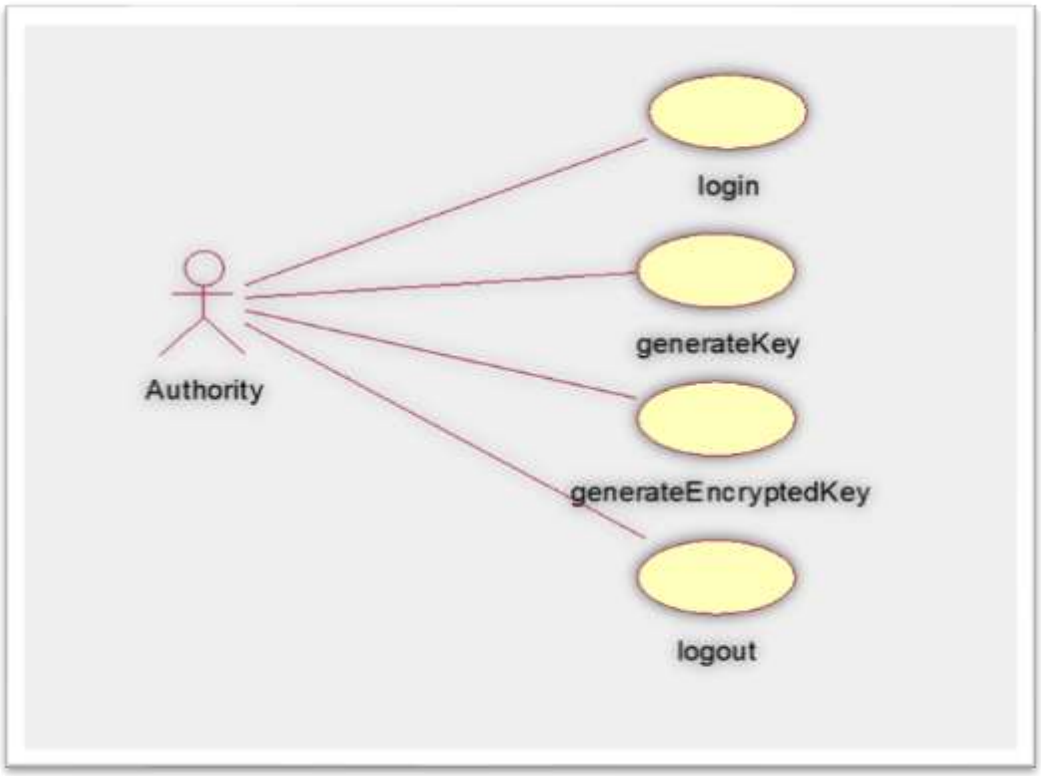


Figure 6.6: use case diagram for Authority

6.3.2 CLASS DIAGRAM

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, operations (or methods), and the relationships among the classes.

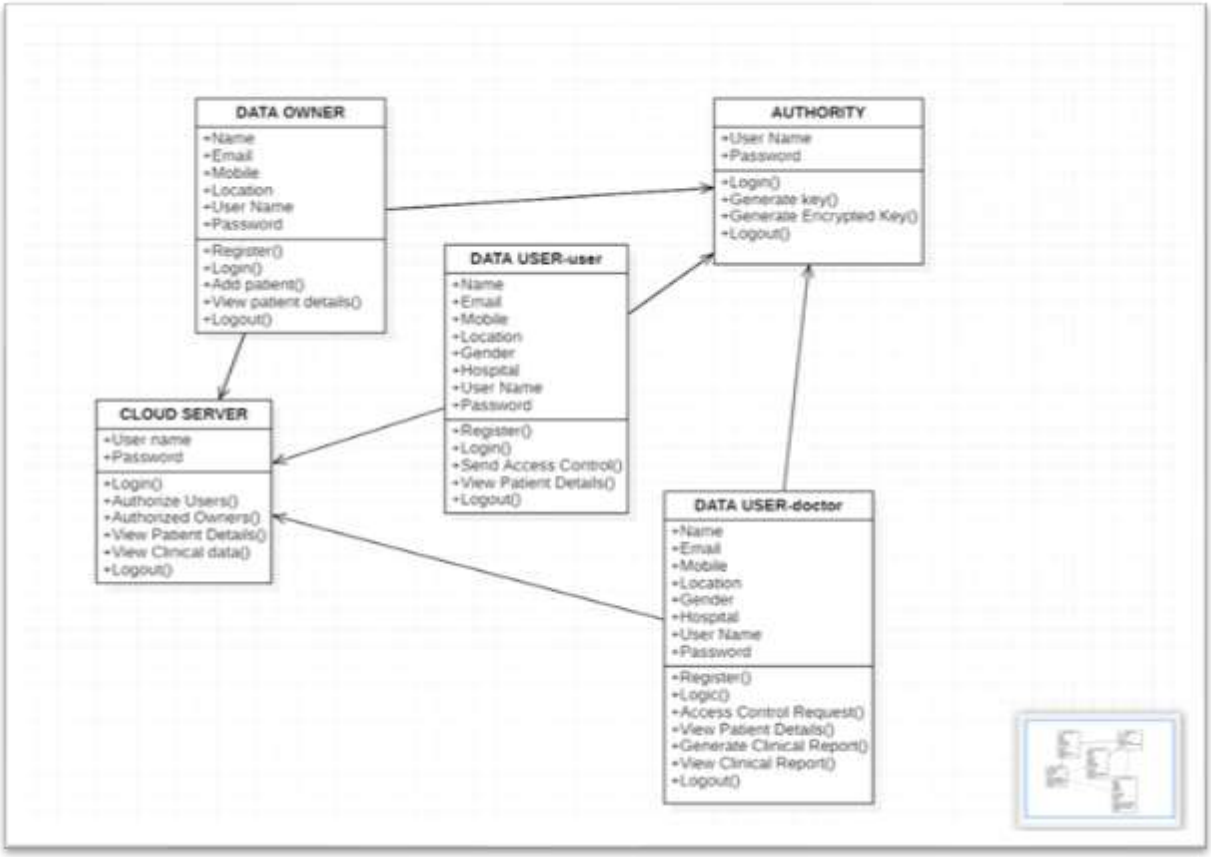


Figure 6.7: Class Diagram

6.3.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

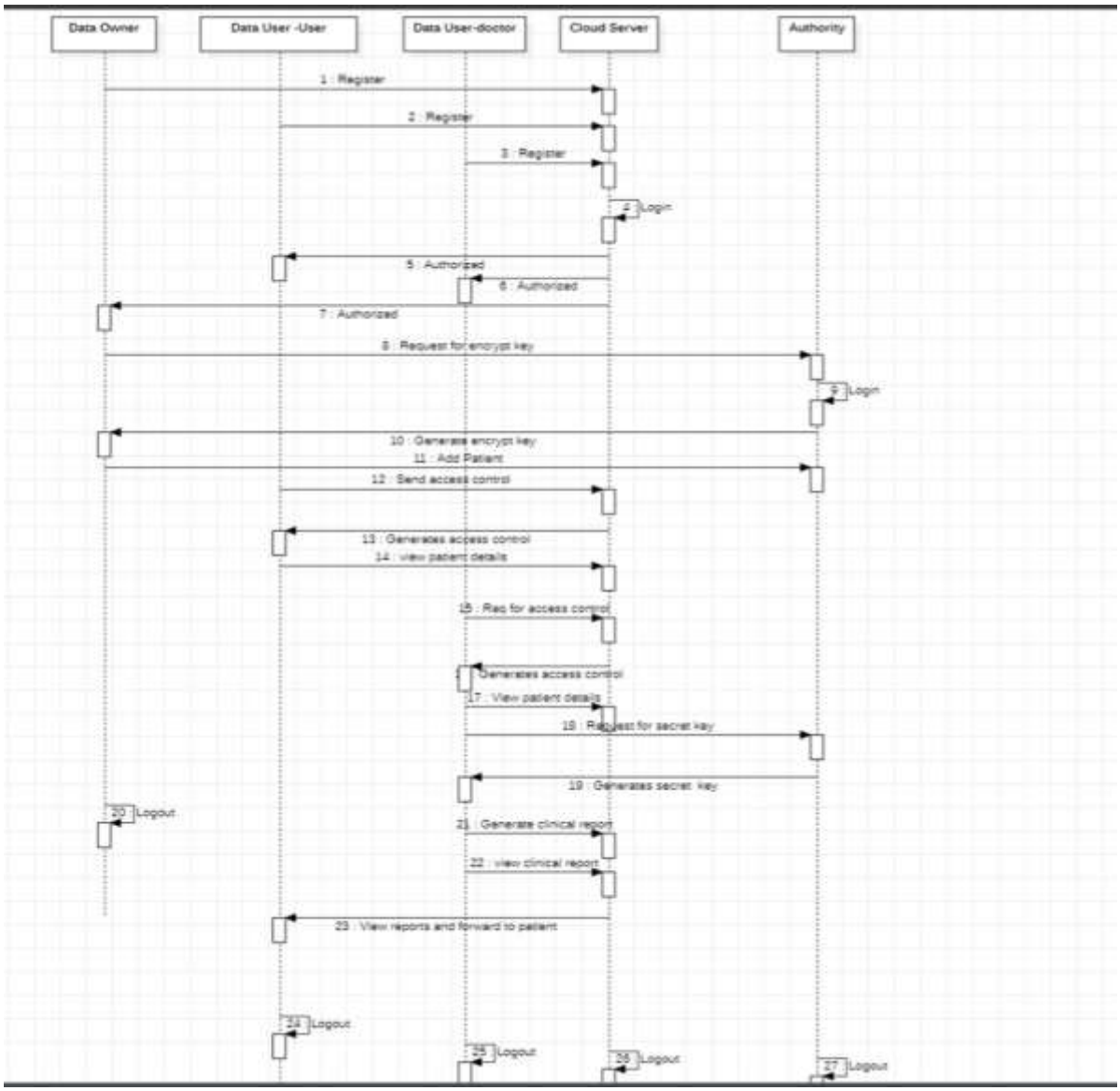


Figure 6.8: Sequence Diagram

CHAPTER 7

SOURCE CODE

DataOwner.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Hidden Ciphertext</title>
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<link rel="stylesheet" href="likeanddislike.css" type="text/css" />
    </head>
<body  id="top">
<div class="wrapper col1">
<div id="head">
<h1><a href="index.html">Hidden Ciphertext Policy </a></h1>
<p>Attribute-Based Encryption with fast decryption for Personal Health Record System</p>
<div id="topnav">
<ul>
<li><a href="index.html">Home</a></li>
<li><a  class="active" href="DataOwner.jsp">Data Owner</a></li>
<li><a href="DataUser.jsp">Data User</a></li>
<li><a href="CloudServer.jsp">Cloud Server</a></li>
<li  class="last"><a href="Authority.jsp">Authority</a></li>
    </ul>
     </div>
   </div>
 </div>
```

```
<div class="wrapper col2">

<div id="gallery">

<ul>

<li class="placeholder" style="background-image:url(images/demo/gallery_default.jpg);">Image Holder</li>

 <li><a class="swap" style="background-image:url(images/slide1.jpg); background-size:290px 105px;" href="#gallery"><strong>Services</strong><span><img src="images/slide1.jpg" alt="" width="640" height="349"/></span></a></li>

  <li><a class="swap" style="background-image:url(images/slide2.jpg);  background-size:290px 105px;" href="#gallery"><strong>Products</strong><span><img src="images/slide2.jpg" alt="" width="640" height="349"/></span></a></li>

  <li class="last"><a class="swap" style="background-image:url(images/slide3.jpg); background-size:290px 105px;" href="#gallery"><strong>Company</strong><span><img src="images/slide3.jpg" alt="" width="640" height="349"/></span></a></li>

 </ul>

 <div class="clear"></div>

 </div>

</div>

<div class="wrapper col4">

<div id="container">

 <div id="content">

  <p style="font-family:monospace;font-size:25px;">Owner Login Page<p>

 <%String msg=request.getParameter("msg");

  if(msg!=null){

   if(msg.equalsIgnoreCase("Your not authorized by the Cloud")){

    %>

 <font color="red">Your not authorized by the Cloud</font>

 <%}else{

  %>

   <font color="red">Login Failed Check Username/password</font>

   <%}

   }%>
```

```
<form action="ownerauth.jsp" name="reg" method="post" autocomplete="off" enctype="for
data/multipart">
    <table>
<tr><th> UserName :</th><td><input type="text"  name="uname" required></td></tr>
<tr><th> Password :</th><td><input type="password"  name="pwd" required></td></tr>
<tr><td><input type="submit" value="Login">      <input
type="reset"></tD><td>Don't Have An Account? <a href="Register.jsp"><font
color="red">Register</font></a></td></tr>
 </table>
 </form>
 </div>
 <div id="column">
<div id="featured">
<ul>
 <li>
 <h2>Menu</h2>
  <li><a class="active" href="index.html">Home</a></li>
  <li><a href="DataOwner.jsp">Data Owner</a></li>
  <li><a href="DataUser.jsp">Data User</a></li>
  <li class="last"><a href="CloudServer.jsp">Cloud Server</a></li>
  <li  class="last"><a href="Authority.jsp">Authority</a></li>
  </ul>
</div>
</div>
<div class="clear"></div>
</div>
</div>
<div class="wrapper col5">
<div id="footer">
<!-- End Contact Form -->
<!-- End Company Details -->
```

```
<div id="copyright">
    <p class="fl_left">Copyright &copy; 2019 - All Rights Reserved - <a href="#">Domain Name</a></p>
<p class="fl_right">Template by <a target="_blank" href="#" title="kishan">Kishan</a></p>
<br class="clear" />
</div>
<div class="clear"></div>
</div>
</div>
</body>
</html>
```

DataUser.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
 <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Hidden Ciphertext</title>
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<link rel="stylesheet" href="likeanddislike.css" type="text/css" />
 </head>
 <body  id="top">
<div class="wrapper col1">
<div id="head">
 <h1><a href="index.html">Hidden Ciphertext Policy </a></h1>
  <p>Attribute-Based Encryption with fast decryption for Personal Health Record System</p>
  <div id="topnav">
 <ul>
 <li><a href="index.html">Home</a></li>
  <li><a href="DataOwner.jsp">Data Owner</a></li>
```

```
<li><a   class="active" href="DataUser.jsp">Data User</a></li>
<li><a href="CloudServer.jsp">Cloud Server</a></li>
<li  class="last"><a href="Authority.jsp">Authority</a></li>
  </ul>
</div>
</div>
</div>
<div class="wrapper col2">
<div id="gallery">
<ul>
<li class="placeholder" style="background-image:url(images/demo/gallery_default.jpg);">Image Holder</li>
<li><a class="swap" style="background-image:url(images/slide1.jpg); background-size:290px 105px;" href="#gallery"><strong>Services</strong><span><img src="images/slide1.jpg" alt="" width="640" height="349"/></span></a></li>
<li><a class="swap" style="background-image:url(images/slide2.jpg);  background-size:290px 105px;" href="#gallery"><strong>Products</strong><span><img src="images/slide2.jpg" alt="" width="640" height="349"/></span></a></li>
<li class="last"><a class="swap" style="background-image:url(images/slide3.jpg); background-size:290px 105px;" href="#gallery"><strong>Company</strong><span><img src="images/slide3.jpg" alt="" width="640" height="349"/></span></a></li>
</ul>
<div class="clear"></div>
</div>
</div>
<div class="wrapper col4">
<div id="container">
<div id="content">
<p style="font-family:monospace;font-size:25px;">User Login Page<p>
<%String msg=request.getParameter("msg");
  if(msg!=null){
if(msg.equalsIgnoreCase("Your not authorized by the Admin")){
  %>
```

```
 <font color="red">Your not authorized by the Cloud</font>
 <%}else{
%>
 <font color="red">Login Failed Check Username/password</font>
  <%
}
}%>
<form action="U_Action.jsp" name="reg" method="post" autocomplete="off" enctype="form-data/multipart">
 <table>
 <tr><th> UserName :</th><td><input type="text"  name="uname" required></td></tr>
 <tr><th> Password :</th><td><input type="password"  name="pwd" required></td></tr>
 <tr><td><input type="submit" value="Login">  
 <input type="reset"></tD><td>Don't Have An Account? <a href="UserRegister.jsp"><font color="red">Register</font></a></td></tr>
  </table>
 </form>
      </div>
    <div id="column">
      <div id="featured">
     <ul>
       <li>
         <h2>Menu</h2>
      <li><a class="active" href="index.html">Home</a></li>
     <li><a href="DataOwner.jsp">Data Owner</a></li>
     <li><a href="DataUser.jsp">Data User</a></li>
     <li class="last"><a href="CloudServer.jsp">Cloud Server</a></li>
     <li  class="last"><a href="Authority.jsp">Authority</a></li>
     </ul>
      </div>
    </div>
```

```
 <div class="clear"></div>
</div>
</div>
<div class="wrapper col5">
<div id="footer">
  <!-- End Contact Form -->
  <!-- End Company Details -->
  <div id="copyright">
      <p class="fl_left">Copyright &copy; 2019 - All Rights Reserved - <a href="#">Domain Name</a></p>
    <p class="fl_right">Template by <a target="_blank" href="#" title="kishan">Kishan</a></p>
    <br class="clear" />
  </div>
  <div class="clear"></div>
 </div>
</div>
</body>
</html>
```

CloudServer.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
 <head>
      <meta charset="UTF-8">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <title>Hidden Ciphertext</title>
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<link rel="stylesheet" href="likeanddislike.css" type="text/css" />
 </head>
 <body  id="top">
```

```
<div class="wrapper col1">
<div id="head">
<h1><a href="index.html">Hidden Ciphertext Policy </a></h1>
<p>Attribute-Based Encryption with fast decryption for Personal Health Record System</p>
<div id="topnav">
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="DataOwner.jsp">Data Owner</a></li>
<li><a href="DataUser.jsp">Data User</a></li>
<li><a class="active" href="CloudServer.jsp">Cloud Server</a></li>
<li class="last"><a href="Authority.jsp">Authority</a></li>
</ul>
</div>
</div>
</div>
<div class="wrapper col2">
<div id="gallery">
<ul>
<li class="placeholder" style="background-image:url(images/demo/gallery_default.jpg);">Image Holder</li>
<li><a class="swap" style="background-image:url(images/slide1.jpg); background-size:290px 105px;" href="#gallery"><strong>Services</strong><span><img src="images/slide1.jpg" alt="" width="640" height="349"/></span></a></li>
<li><a class="swap" style="background-image:url(images/slide2.jpg); background-size:290px 105px;" href="#gallery"><strong>Products</strong><span><img src="images/slide2.jpg" alt="" width="640" height="349"/></span></a></li>
<li class="last"><a class="swap" style="background-image:url(images/slide3.jpg); background-size:290px 105px;" href="#gallery"><strong>Company</strong><span><img src="images/slide3.jpg" alt="" width="640" height="349"/></span></a></li>
</ul>
<div class="clear"></div>
</div>
</div>
```

```
 <div class="wrapper col4">
  <div id="container">
 <div id="content">
 <p style="font-family:monospace;font-size:25px;">Cloud Login Page<p>
  <%String msg=request.getParameter("msg");
  if(msg!=null){
     %>
<font color="red">Login Failed Check Username/password</font>
  <%
   }%>
  <form action="Cloud_Action.jsp" name="reg" method="post" autocomplete="off"
enctype="form-data/multipart">
       <table>
 <tr><th> UserName :</th><td><input type="text"  name="uname" required></td></tr>
  <tr><th> Password :</th><td><input type="password"  name="pwd" required></td></tr>
   <tr><td><input type="submit" value="Login">   </td>
   <td><input type="reset" value="Reset">   </td></tr>
</table>
 </form>
  </div>
 <div id="column">
 <div id="featured">
<ul>
 <li>
 <h2>Menu</h2>
 <li><a class="active" href="index.html">Home</a></li>
 <li><a href="DataOwner.jsp">Data Owner</a></li>
 <li><a href="DataUser.jsp">Data User</a></li>
 <li class="last"><a href="CloudServer.jsp">Cloud Server</a></li>
 <li  class="last"><a href="Authority.jsp">Authority</a></li>
</ul>
```

```
</div>
</div>
 <div class="clear"></div>
</div>
</div>
<div class="wrapper col5">
<div id="footer">
<!-- End Contact Form -->
<!-- End Company Details -->
<div id="copyright">
<p class="fl_left">Copyright &copy; 2019 - All Rights Reserved - <a href="#">Domain Name</a></p>
<p class="fl_right">Template by <a target="_blank" href="#" title="kishan">Kishan</a></p>
<br class="clear" />
</div>
<div class="clear"></div>
</div>
 </div>
</body>
</html>
```

Authority.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Hidden Ciphertext</title>
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
<link rel="stylesheet" href="likeanddislike.css" type="text/css" />
```

```
</head>
<body  id="top">
  <div class="wrapper col1">
 <div id="head">
 <h1><a href="index.html">Hidden Ciphertext Policy </a></h1>
   <p>Attribute-Based Encryption with fast decryption for Personal Health Record System</p>
   <div id="topnav">
   <ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="DataOwner.jsp">Data Owner</a></li>
  <li><a  href="DataUser.jsp">Data User</a></li>
  <li><a href="CloudServer.jsp">Cloud Server</a></li>
  <li  class="last"><a    class="active" href="Authority.jsp">Authority</a></li>
    </ul>
    </div>
  </div>
  </div>
 <div class="wrapper col2">
<div id="gallery">
  <ul>
  <li class="placeholder" style="background-image:url(images/demo/gallery_default.jpg);">Image Holder</li>
  <li><a class="swap" style="background-image:url(images/slide1.jpg); background-size:290px 105px;" href="#gallery"><strong>Services</strong><span><img src="images/slide1.jpg" alt="" width="640" height="349"/></span></a></li>
   <li><a class="swap" style="background-image:url(images/slide2.jpg);  background-size:290px 105px;" href="#gallery"><strong>Products</strong><span><img src="images/slide2.jpg" alt="" width="640" height="349"/></span></a></li>
   <li class="last"><a class="swap" style="background-image:url(images/slide3.jpg); background-size:290px 105px;" href="#gallery"><strong>Company</strong><span><img src="images/slide3.jpg" alt="" width="640" height="349"/></span></a></li>
  </ul>
  <div class="clear"></div>
```

```
 </div>
 </div>
 <div class="wrapper col4">
 <div id="container">
 <div id="content">
   <p style="font-family:monospace;font-size:25px;">Authority Login Page<p>
   <%String msg=request.getParameter("msg");
  if(msg!=null){
%>
            <font color="red">Login Failed Check Username/password</font>
  <%
 }%>
 <form action="Authority_Action.jsp" name="reg" method="post" autocomplete="off"
enctype="form-data/multipart">
 <table>
 <tr><th> UserName :</th><td><input type="text"  name="uname" required></td></tr>
<tr><th> Password :</th><td><input type="password"  name="pwd" required></td></tr>
<tr><td><input type="submit" value="Login">   </td>
<td><input type="reset" value="Reset">   </td></tr>
 </table>
 </form>
  </div>
  <div id="column">
 <div id="featured">
 <ul>
  <li>
   <h2>Menu</h2>
   <li><a class="active" href="index.html">Home</a></li>
  <li><a href="DataOwner.jsp">Data Owner</a></li>
  <li><a href="DataUser.jsp">Data User</a></li>
   <li class="last"><a href="CloudServer.jsp">Cloud Server</a></li>
```

```
<li  class="last"><a href="Authority.jsp">Authority</a></li>
</ul>
</div>
</div>
<div class="clear"></div>
</div>
</div>
<div class="wrapper col5">
<div id="footer">
<!-- End Contact Form -->
<!-- End Company Details -->
<div id="copyright">
<p class="fl_left">Copyright &copy; 2019 - All Rights Reserved - <a href="#">Domain
Name</a></p>
<p class="fl_right">Template by <a target="_blank" href="#" title="kishan">Kishan</a></p>
<br class="clear" />
</div>
<div class="clear"></div>
</div>
</div>
</body>
</html>
```

Index.html

```
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Hidden Ciphertext</title>
<link rel="stylesheet" href="layout/styles/layout.css" type="text/css" />
```

```
</head>
<body  id="top">
 <div class="wrapper col1">
  <div id="head" style="width:1300px;">
  <h1 style="float:left;"><a href="index.html">Hidden Ciphertext Policy </a></h1>
  <p>Attribute-Based Encryption with fast decryption for Personal Health Record System</p>
   <div id="topnav">
     <ul >
   <li><a class="active" href="index.html">Home</a></li>
   <li><a href="DataOwner.jsp">Data Owner</a></li>
   <li><a href="DataUser.jsp">Data User</a></li>
   <li><a href="CloudServer.jsp">Cloud Server</a></li>
   <li  class="last"><a href="Authority.jsp">Authority</a></li>
     </ul>
    </div>
    </div>
   </div>
    <div class="wrapper col2">
  <div id="gallery">
   <ul>
        <li class="placeholder" style="background-
  image:url(images/gallery_default.jpg);background-size: 650px 400px;">Image Holder</li>
       <li><a class="swap" style="background-image:url(images/slide1.jpg); background-size:290px
  105px;" href="#gallery"><strong>Services</strong><span><img src="images/slide1.jpg" alt=""
  width="640" height="349"/></span></a></li>
       <li><a class="swap" style="background-image:url(images/slide2.jpg);  background-size:290px
  105px;" href="#gallery"><strong>Products</strong><span><img src="images/slide2.jpg" alt=""
  width="640" height="349"/></span></a></li>
       <li class="last"><a class="swap" style="background-image:url(images/slide3.jpg);
  background-size:290px 105px;" href="#gallery"><strong>Company</strong><span><img
  src="images/slide3.jpg" alt="" width="640" height="349"/></span></a></li>
  </ul>
   <div class="clear"></div>
```

```
 </div>
</div>
<div class="wrapper col4">
<div id="container">
 <div id="content">
  <h1>About The Abstract</h1>
  <p align="justify">
```

We propose two new ciphertext policy attribute based encryption (CP-ABE) schemes where the access policy is defined by AND-gate with wildcard. In the first scheme,we present a new technique that uses only one group element to represent an attribute, while the existing ABE schemes of the same type need to use three different group elements to represent an attribute for the three possible values (namely,positive, negative, and wildcard). Our new technique leads to a new CP-ABE scheme with constant ciphertext size, which,however, cannot hide the access policy used for encryption. The main contribution of this paper is to propose a new CP-ABE scheme with the property of hidden access policy by extendingthe technique we used in the construction of our first scheme.In particular, we show a way to bridge ABE based on AND-gate with wildcard with inner product encryption and then use the latter to achieve the goal of hidden access policy. We prove that our second scheme is secure under the standard decisional linearand decisional bilinear Diffie Hellman assumptions.

```
  </p>
</div>
<div id="column">
<div id="featured">
 <ul>
  <li>
   <h2>Menu</h2>
    <li><a class="active" href="index.html">Home</a></li>
 <li><a href="DataOwner.jsp">Data Owner</a></li>
 <li><a href="DataUser.jsp">Data User</a></li>
 <li class="last"><a href="CloudServer.jsp">Cloud Server</a></li>
 <li  class="last"><a href="Authority.jsp">Authority</a></li>
 </ul>
</div>
</div>
<div class="clear"></div>
```

```
</div>
</div>
  <div class="wrapper col5">
<div id="footer">
<!-- End Contact Form -->
<!-- End Company Details -->
 <div id="copyright">
  <p class="fl_left">Copyright &copy; 2019 - All Rights Reserved - <a href="#">Domain Name</a></p>
  <p class="fl_right">Template by <a target="_blank" href="#" title="kishan">Kishan</a></p>
  <br class="clear" />
  </div>
   <div class="clear"></div>
 </div>
 </div>
 </body>
  </html>
```

RegisterServlet.java

```
package com.registration;

import com.dbcon.DBCon;
import com.dbcon.Queries;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.InputStream;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
```

```
import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;

@MultipartConfig(maxFileSize=16177215)
public class RegisterServlet extends HttpServlet {


    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        try{


String name=request.getParameter("name");
String email=request.getParameter("email");
String mobile=request.getParameter("mobile");
String address=request.getParameter("address");
String uname=request.getParameter("uname");
String pass=request.getParameter("pwd");

String location=request.getParameter("location");
  Part image=request.getPart("image");
  InputStream inputstream=null;
  if(image!=null){
     inputstream=image.getInputStream();
```

```
 }

        Connection con=DBCon.getCon();
        String query="select count(*) from ownerreg where email='"+email+"'";
        ResultSet r=Queries.getExecuteQuery(query);
        int count=0;
         while(r.next()){
            count=r.getInt(1);
            if(count==0){
PreparedStatement st=con.prepareStatement("insert into ownerreg values(null,?,?,?,?,?,?,?,?,?)");

st.setString(1,name);
st.setString(2,email);
st.setString(3,mobile);
st.setString(4,address);
st.setString(5,uname);
st.setString(6,pass);
st.setString(7,location);

 if(inputstream!=null){
 st.setBlob(8,inputstream);
  }
 st.setString(9,"waiting");

 int i=st.executeUpdate();
if(i>0){
response.sendRedirect("Register.jsp?msg=success");
}else{
  response.sendRedirect("Register.jsp?msg=failed");
  }
```

```
    }else{
    response.sendRedirect("Register.jsp?msg=EMAIL ALREADY EXIST");
    }
      }
   }catch(Exception e){
     out.println(e);
   }
    }
   // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left
 to edit    the code.">
 /**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
       throws ServletException, IOException {
   processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
```

```
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>
}
```

Decryption.java

```
import com.sun.org.apache.xerces.internal.impl.dv.util.Base64;
import java.io.ByteArrayOutputStream;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.util.Scanner;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.swing.JOptionPane;
import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;
```

```
public class decryption {
//public static void main(String args[])
//{
//          Scanner s=new Scanner(System.in);
//           System.out.println("Enter encrypted Text and key");
//           String text=s.next();
//           String key=s.next();
//         new decryption().decrypt(text,key);
//}
    public String decrypt(String txt, String skey) {
        String decryptedtext = null;
        try {
            //converting string to secretkey
            byte[] bs = Base64.decode(skey);
            SecretKey sec = new SecretKeySpec(bs, "AES");
            System.out.println("converted string to seretkey:" + sec);
            System.out.println("secret key:" + sec);
            Cipher aesCipher = Cipher.getInstance("AES");//getting AES instance
            aesCipher.init(Cipher.ENCRYPT_MODE, sec);//initiating ciper encryption using secretkey
            byte[] byteCipherText = new BASE64Decoder().decodeBuffer(txt); //encrypting data
            //  System.out.println("ciper text:"+byteCipherText);
             aesCipher.init(Cipher.DECRYPT_MODE, sec, aesCipher.getParameters());//initiating ciper
            decryption
            byte[] byteDecryptedText = aesCipher.doFinal(byteCipherText);
            decryptedtext = new String(byteDecryptedText);
            System.out.println("Decrypted Text:" + decryptedtext);
        } catch (Exception e) {
            System.out.println(e);
        }
        return decryptedtext;
      }
```

```
}
```

Encryption.java

```
package com.upload;

import com.sun.org.apache.xerces.internal.impl.dv.util.Base64;
import java.io.ByteArrayOutputStream;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.util.Scanner;

import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.swing.JOptionPane;
import sun.misc.BASE64Encoder;

public class encryption {

   public String encrypt(String text, SecretKey secretkey) {
      String plainData = null, cipherText = null;
      try {
         plainData = text;
         Cipher aesCipher = Cipher.getInstance("AES");//getting AES instance
            aesCipher.init(Cipher.ENCRYPT_MODE, secretkey);//initiating ciper encryption using
   secretkey
         byte[] byteDataToEncrypt = plainData.getBytes();
         byte[] byteCipherText = aesCipher.doFinal(byteDataToEncrypt);//encrypting data
```

```
 cipherText = new BASE64Encoder().encode(byteCipherText);//converting encrypted data to
string
   System.out.println("\n Given text : " + plainData + " \n Cipher Data : " + cipherText);


} catch (Exception e) {
   System.out.println(e);
}
return cipherText;
}


}
```

Key.java

```
package com.upload;


/*** * @author Acer */
public class Key {


private static final String ALPHA_NUMERIC_STRING =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";
public static String randomAlphaNumeric(int count) {
StringBuilder builder = new StringBuilder();
while (count-- != 0) {
int character = (int)(Math.random()*ALPHA_NUMERIC_STRING.length());
builder.append(ALPHA_NUMERIC_STRING.charAt(character));
}
return builder.toString();
}
}
```

DBcon.java

```
package com.dbcon;
import java.sql.*;
/**
 *
 * @author Acer
 */
public class DBCon {
    public static Connection con=null;
    public static Connection getCon(){
    try{
    Class.forName("com.mysql.jdbc.Driver");
 con=DriverManager.getConnection("jdbc:mysql://localhost:3306/Hidden_Ciphertext","root","root");

}catch(Exception e){
    System.out.println(e);

  }
    return con;
    }
}
```

CHAPTER 8

SYSTEM TESTING

8.1 TESTING METHODOLOGIES

The motivation behind testing is to seek out blunders. Testing is that the approach toward making an attempt to seek out every attainable blame or disadvantage in an exceedingly work item. It offers associate degree approach to examine the utility of segments, sub congregations, gatherings or doubtless a completed item it's the approach toward active programming with the expectation of guaranteeing that the Programming framework lives up to its stipulations associate degreed consumer wishes and doesn't bomb in a disappointing approach. There are a unit differing types of take a look at. every take a look at compose addresses a selected testing necessity.

8.1.1 UNIT TESTING:

Unit testing is often directed as a significant side of a joined code and unit trial period of the merchandise lifecycle, despite the very fact that it's not extraordinary for writing and unit testing to be junction rectifier as 2 clear stages. Test procedure and approach Field testing are going to be performed physically and utilitarian tests are going to be composed very well.

Test destinations

•All field passages should work lawfully.

•Pages should be enacted from the distinguished association.

•The section screen, messages and reactions should not be deferred. Highlights to be tried

•Verify that the sections area unit of the proper configuration

•No copy passages got to be permissible

•All connections ought to take the consumer to the proper page.

8.1.2 INTEGRATION TESTING

Programming reconciliation testing is that the progressive connexon testing of a minimum of 2 coordinated programming segments on a solitary stage to deliver disappointments caused by interface abandons.

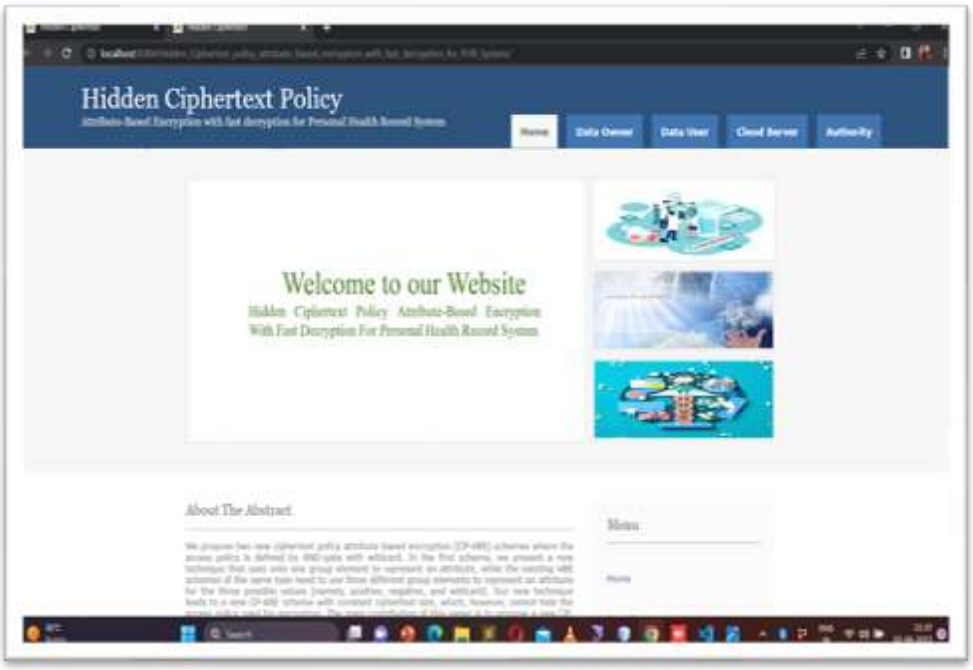
The trip of the incorporation take a look at is to look at that components or programming applications, e.g. segments in an exceedingly product framework or – one stage up – programming applications at the organization level – connect while not mistake. Test outcomes: All the experiments aforementioned higher than passed effectively. No deformities veteran.

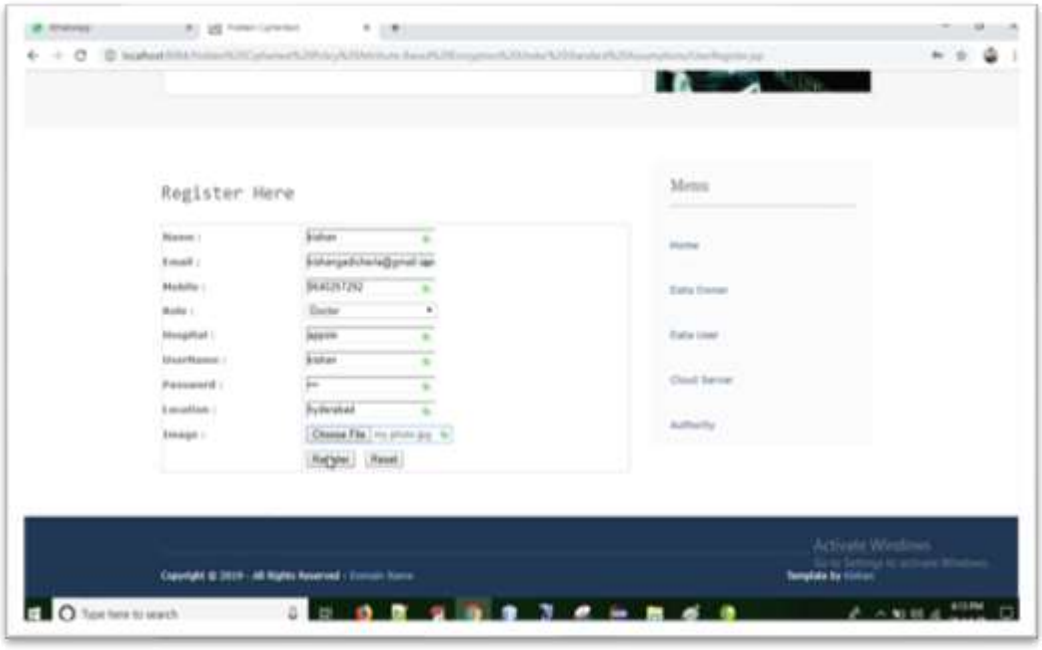
8.1.3 ACCEPTANCE TESTING

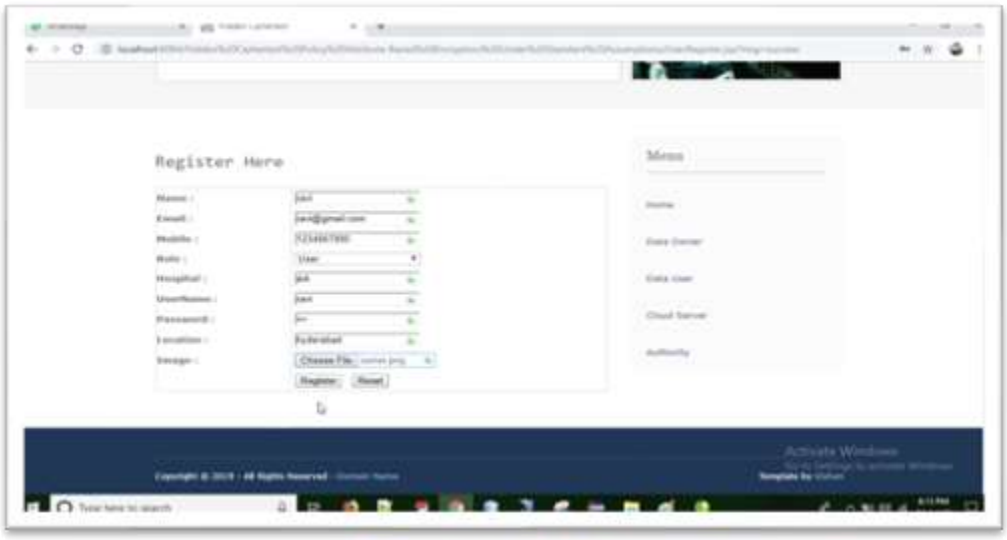
Client Acceptance Testing may be a basic amount of any task and needs Brobdingnagian cooperation by the top consumer. It likewise guarantees that the framework meets the utilitarian stipulations. Test outcomes: All the experiments specified higher than passed effectively. No deformities veteran.

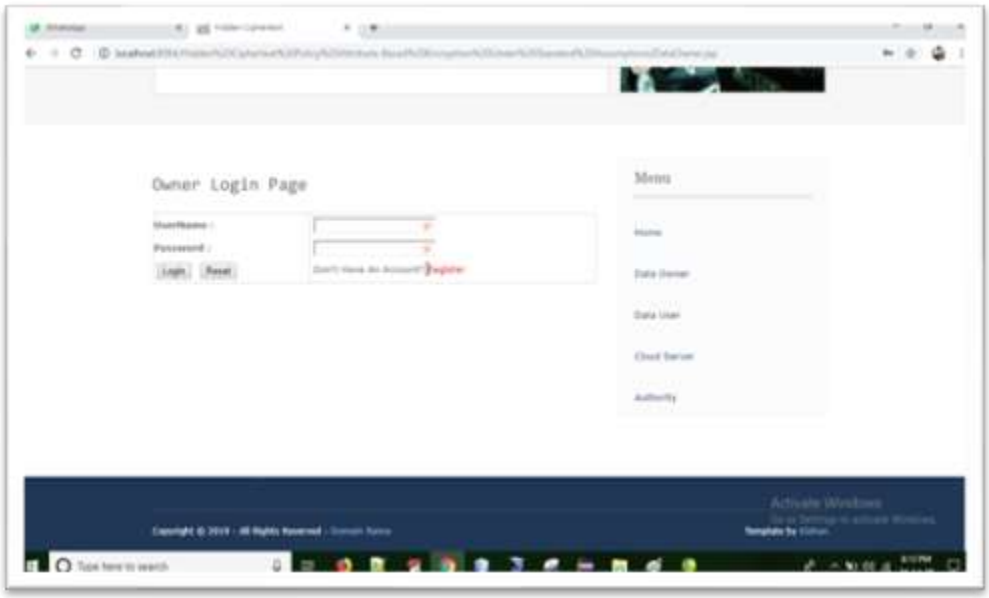
CHAPTER 9

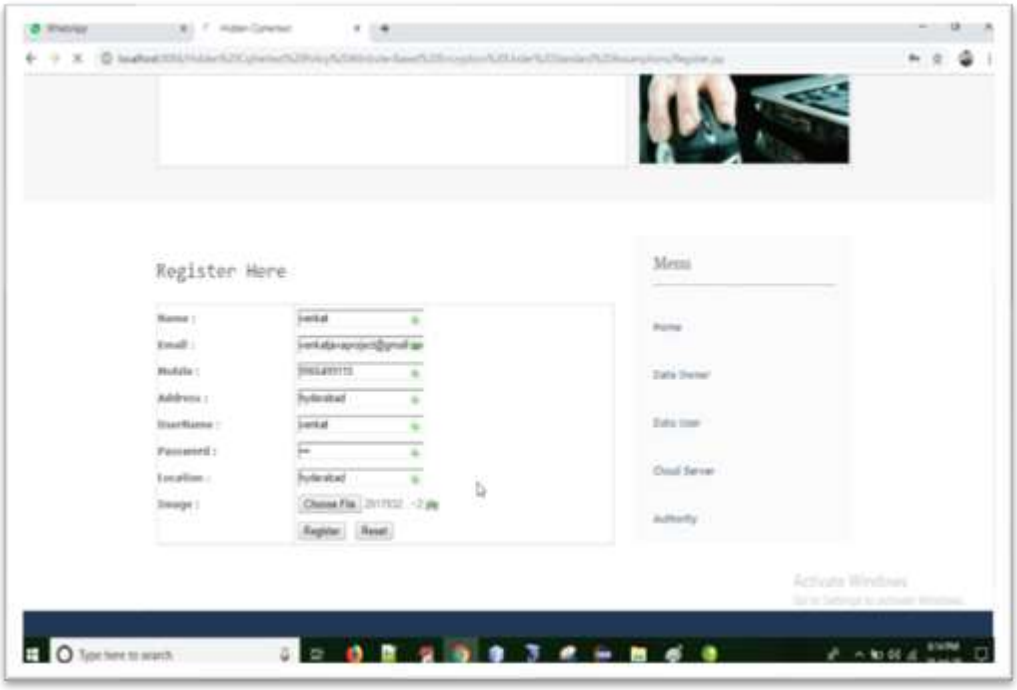
OUTPUT SCREENS

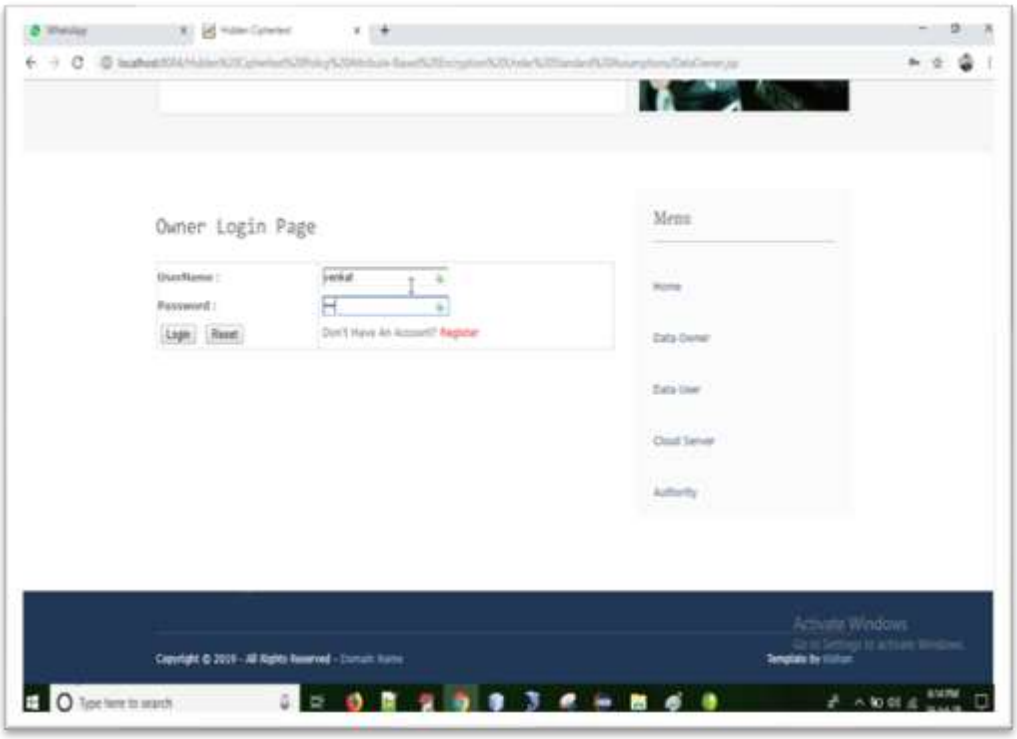


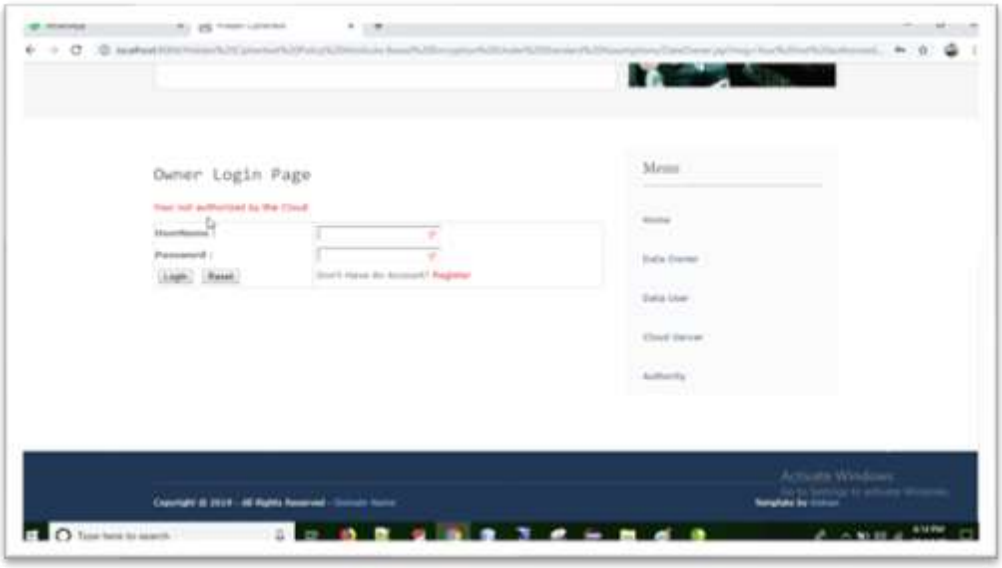


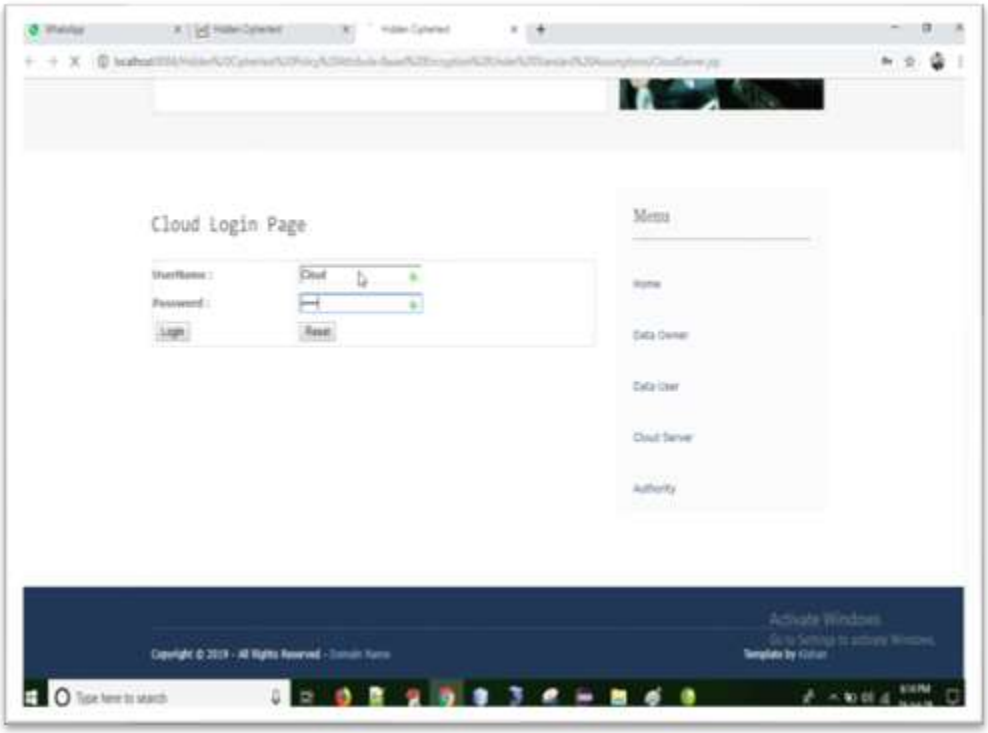

Register Here
Menu
Home
Data Owner
Data User
Cloud Server
Authority
Activate Windows

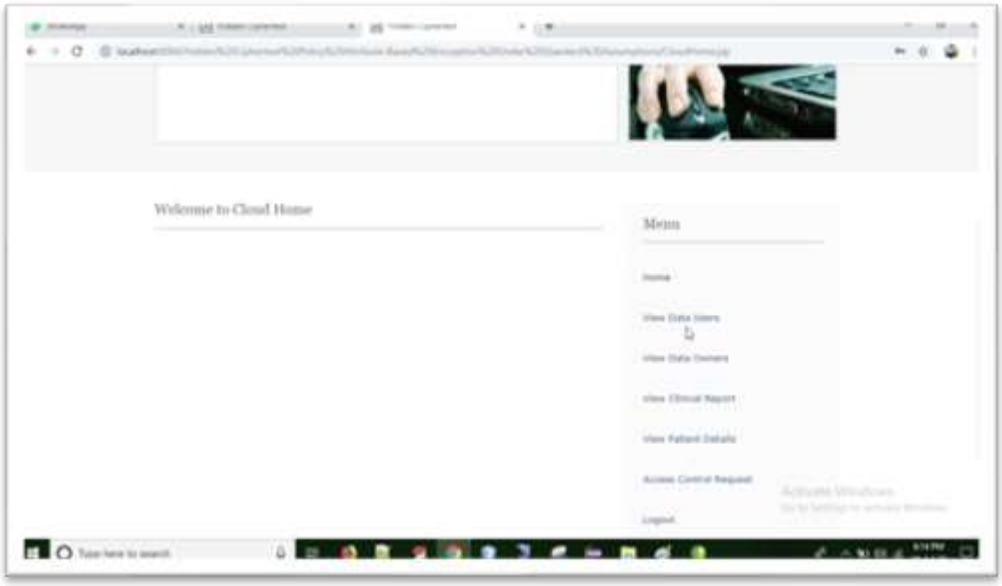

Owner Login Page
Menu
Home
Data Owner
Data User
Cloud Server
Authority
Activate Windows

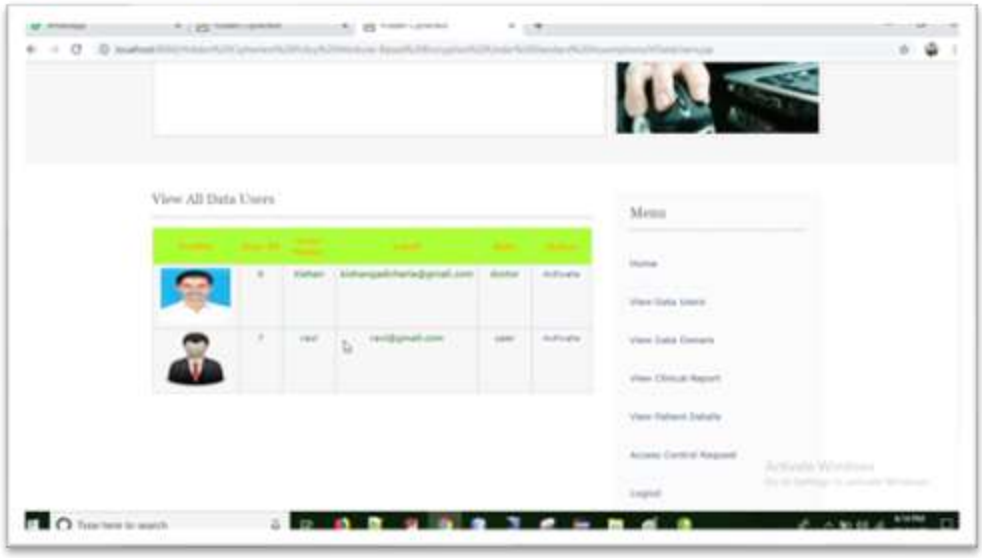

Register Here
Menu
Home
Data Owner
Data User
Cloud Server
Authority
Activate Windows
Type here to search

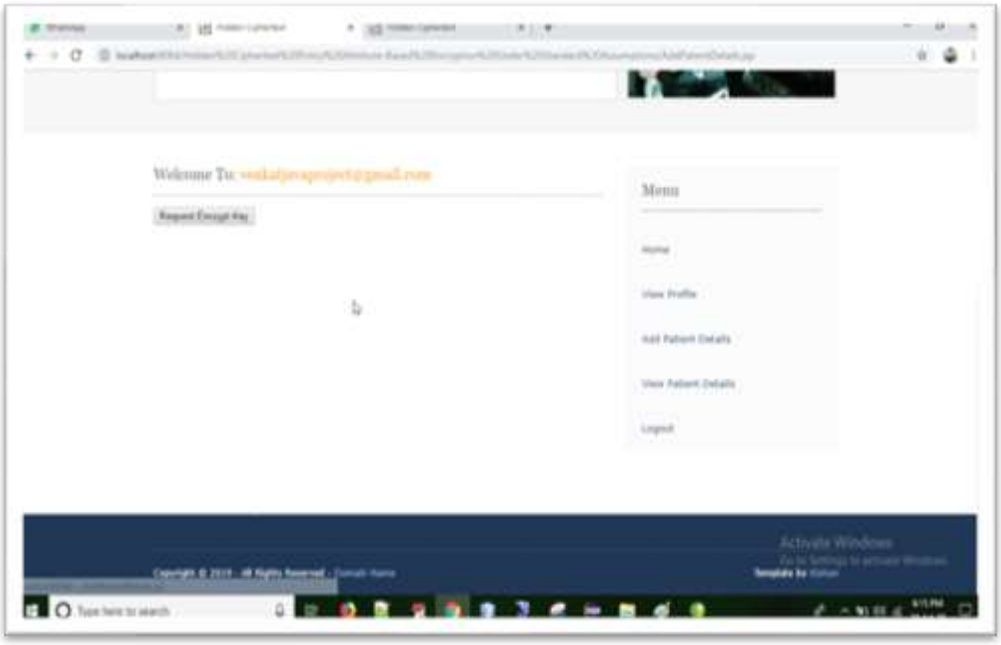

Owner Login Page
UserName :
Password :
Login
Reset
Don't Have An Account? Register
Menu
Home
Data Owner
Data User
Cloud Server
Authority
Activate Windows
Type here to search

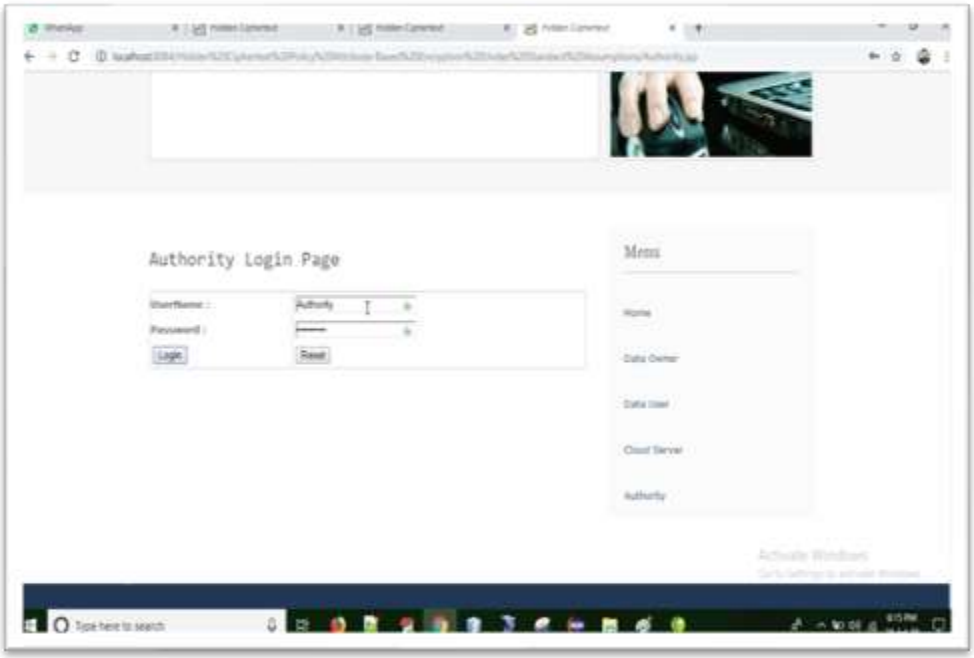

Owner Login Page
Menu
Home
Data Owner
Data User
Cloud Server
Authority

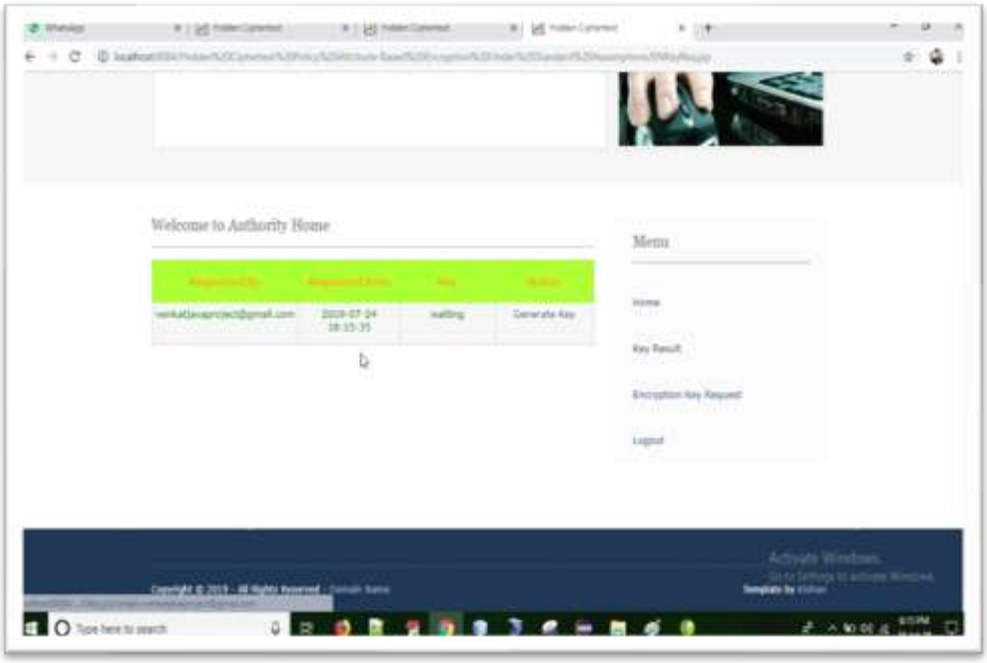

Cloud Login Page
Menu
Home
Data Owner
Data User
Cloud Server
Authority

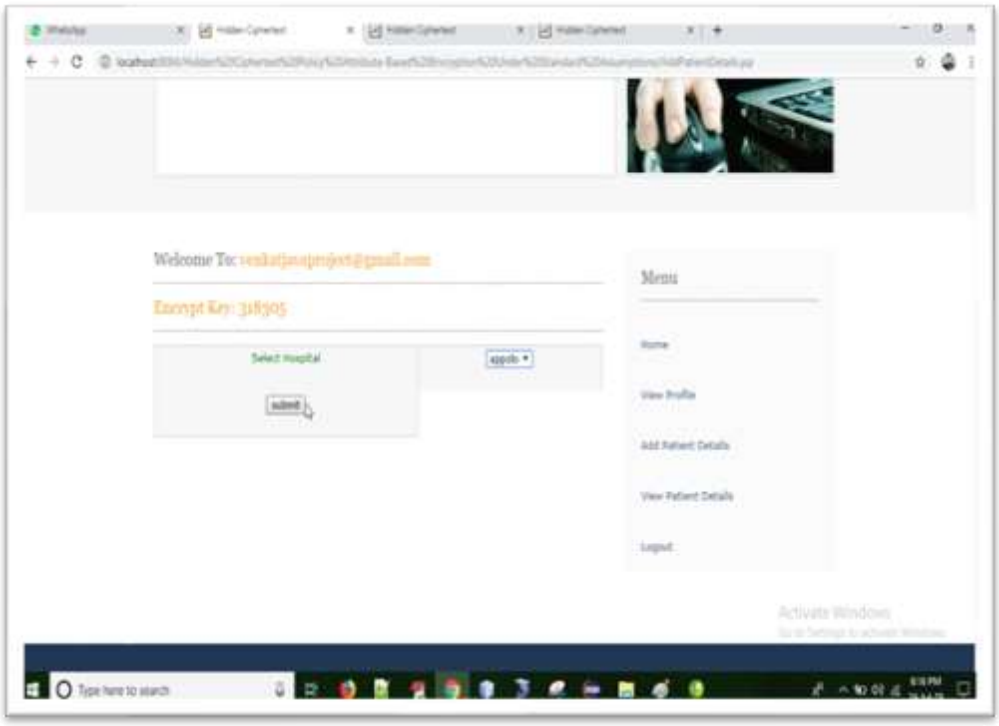

Welcome to Cloud Home
Menu
Home
View Data Users
View Data Owners
view Clinical Report
View Patient Details
Access Control Request
Logout

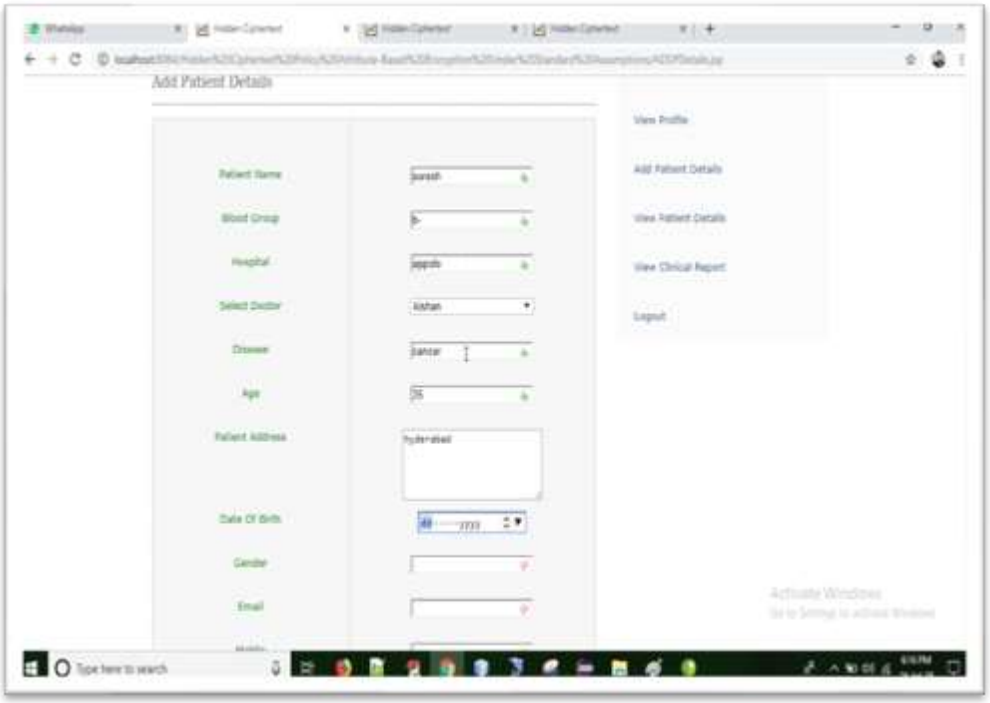

View All Data Users
Menu
Home
View Data Users
View Data Owners
View Clinical Report
View Patient Details
Access Control Request
Logout

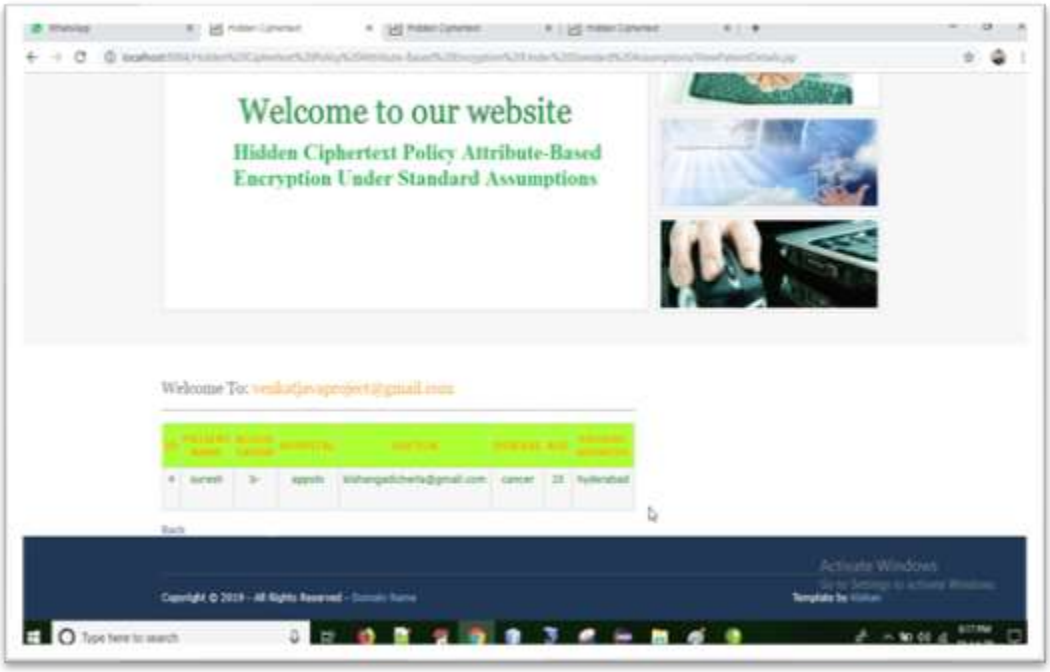

Welcome To
Menu
Home
View Profile
Add Patient Details
View Patient Details
Logout

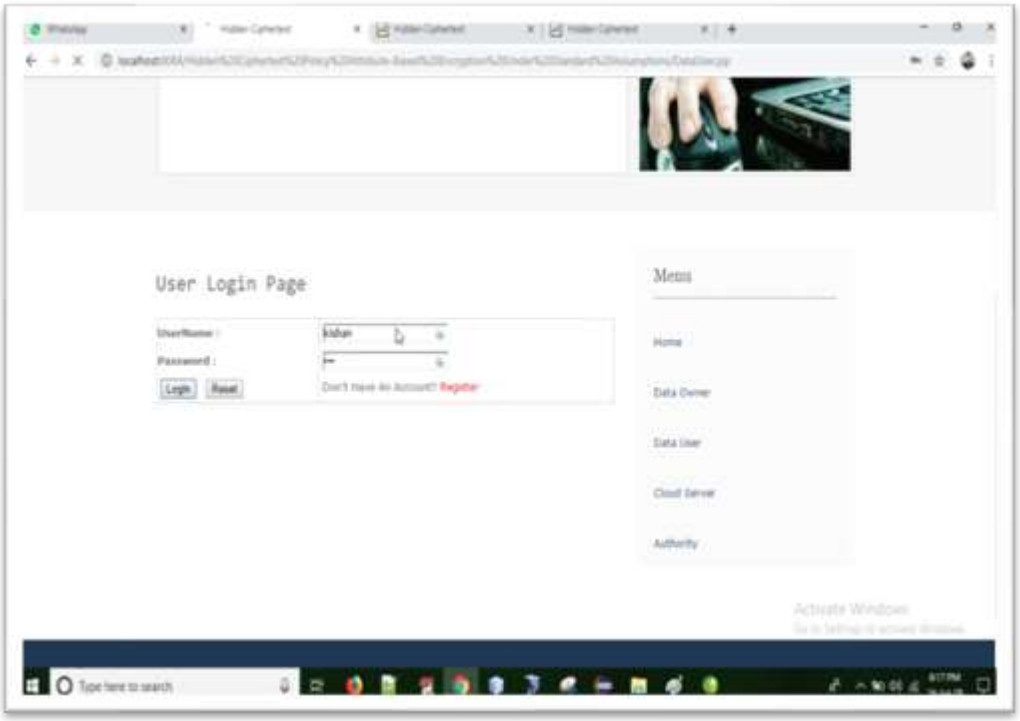

Authority Login Page
Menu
Home
Data Owner
Data User
Cloud Server
Authority


Welcome to Authority Home
Menu
Home
Key Result
Encryption Key Request
Logout


Welcome To:
Encrypt Key: 318305
Select Hospital
apollo
submit
Menu
Home
View Profile
Add Patient Details
View Patient Details
Logout

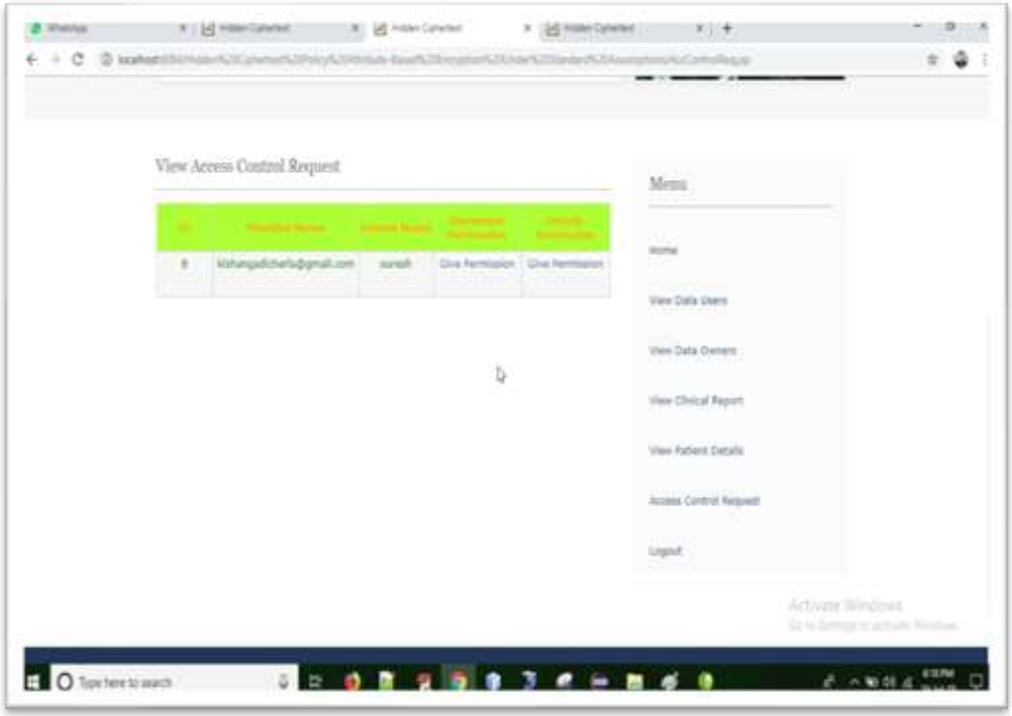

Add Patient Details
View Profile
Add Patient Details
View Patient Details
View Clinical Report
Logout
Patient Name
Blood Group
Hospital
Select Doctor
Disease
Age
Patient Address
Date Of Birth
Gender
Email


Welcome to our website
Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions
Welcome To:

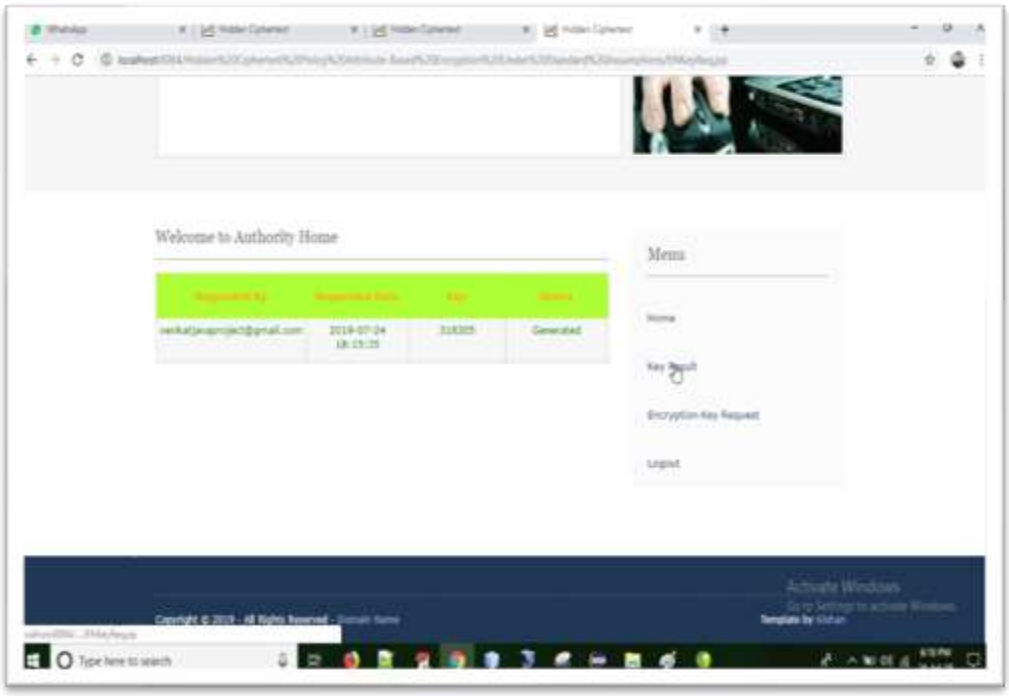

User Login Page
Menu
Home
Data Owner
Data User
Cloud Server
Authority

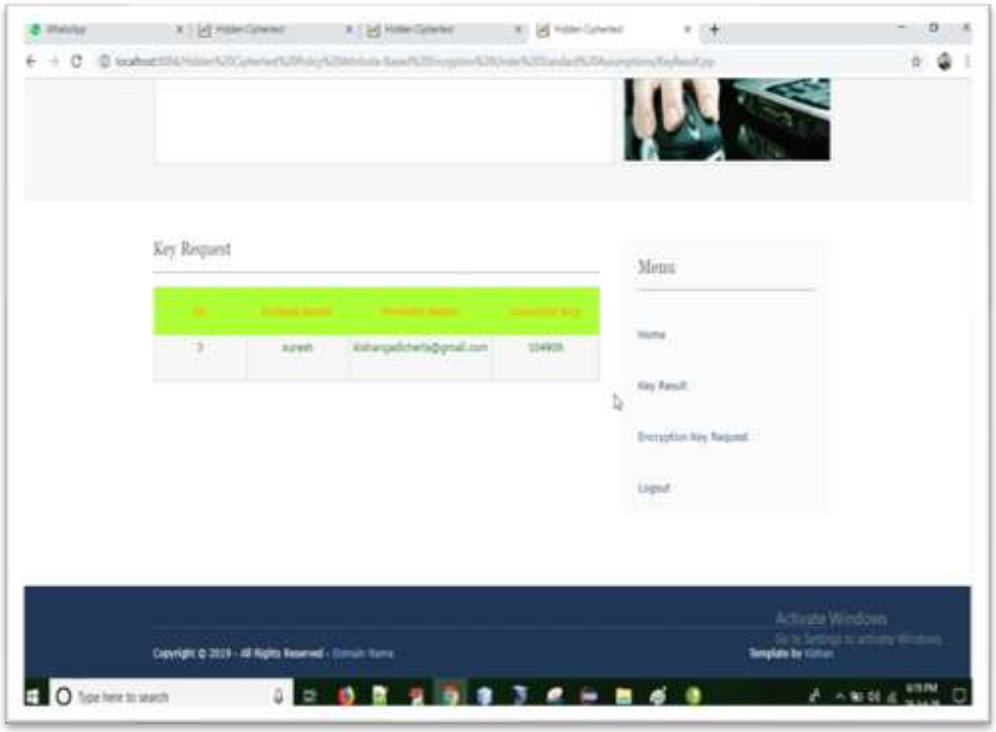

Role: doctor
Back


Encryption Under Standard Assumptions
Welcome To: kishanpadicherla@gmail.com
Role: doctor
Back
Copyright © 2019 - All Rights Reserved
Activate Windows


View Access Control Request
Menu
Home
View Data Users
View Data Owners
View Clinical Report
View Patient Details
Access Control Request
Logout
Activate Windows

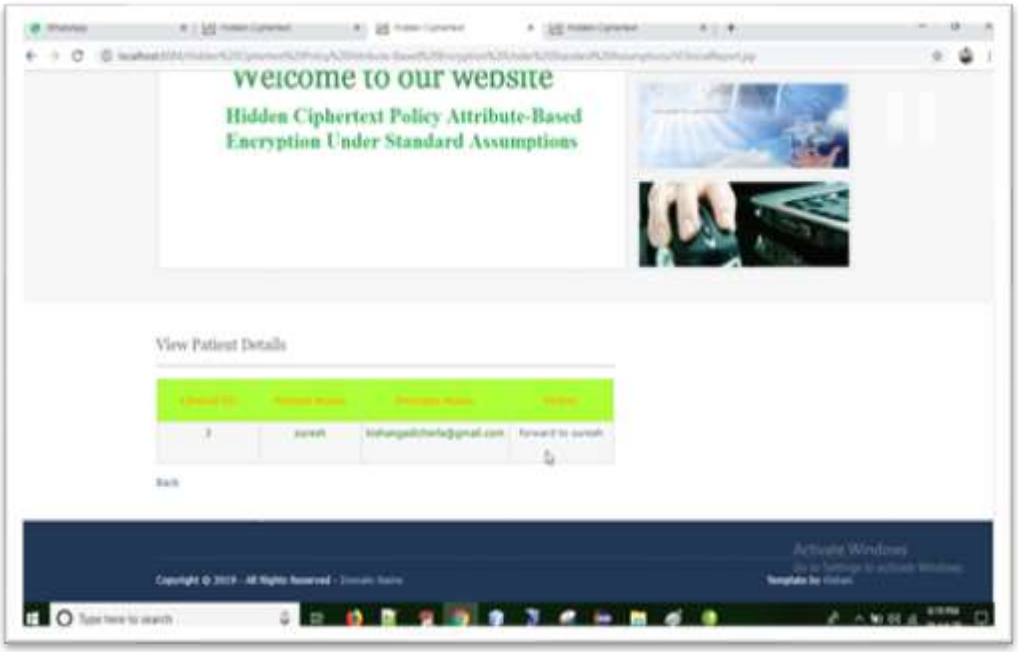

Welcome To
Role: doctor
Generate Clinical Report
suresh
tumor
male
Activate Windows
Type here to search

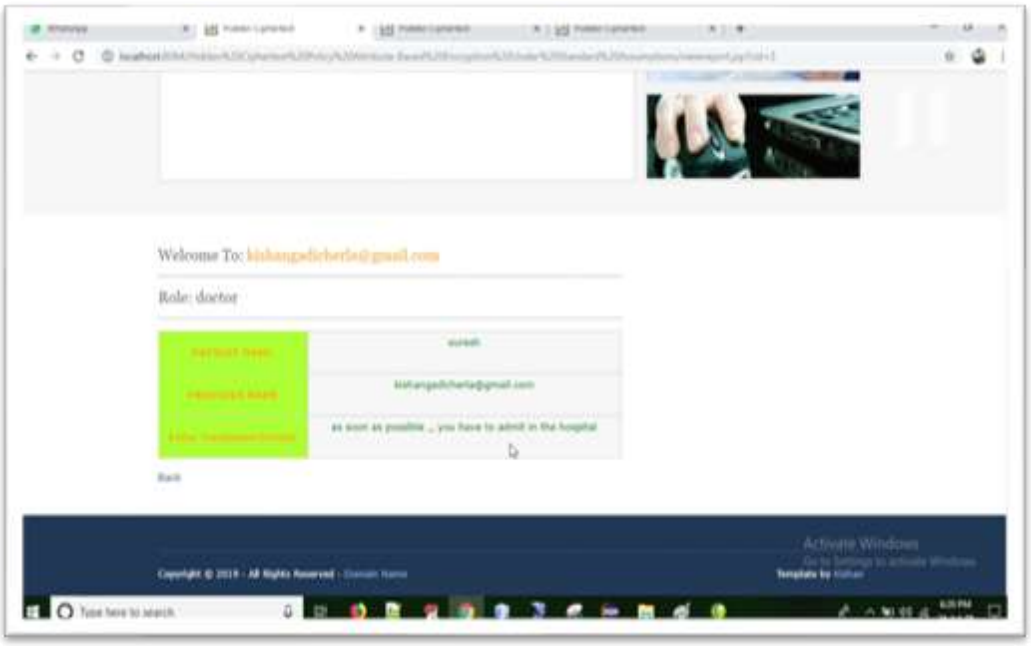

Welcome to Authority Home
Menu
Home
Encryption Key Request
Logout
Activate Windows
Type here to search

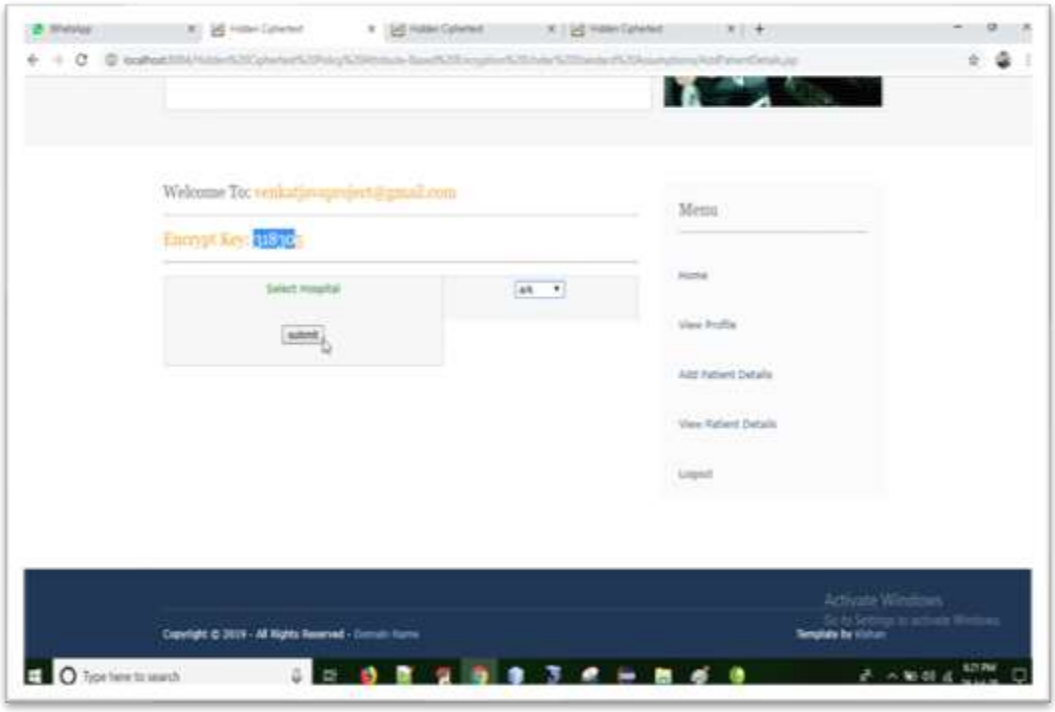

Key Request
Menu
Home
Key Result
Encryption Key Request
Logout
Activate Windows

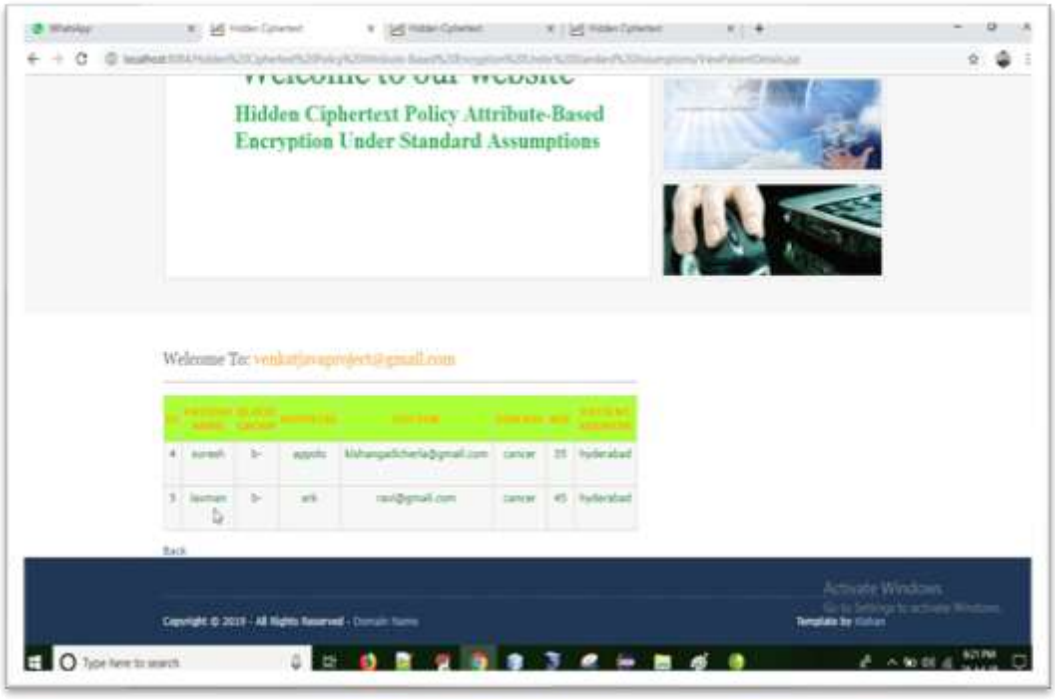

Role: doctor
Generate Clinical Report
Activate Windows

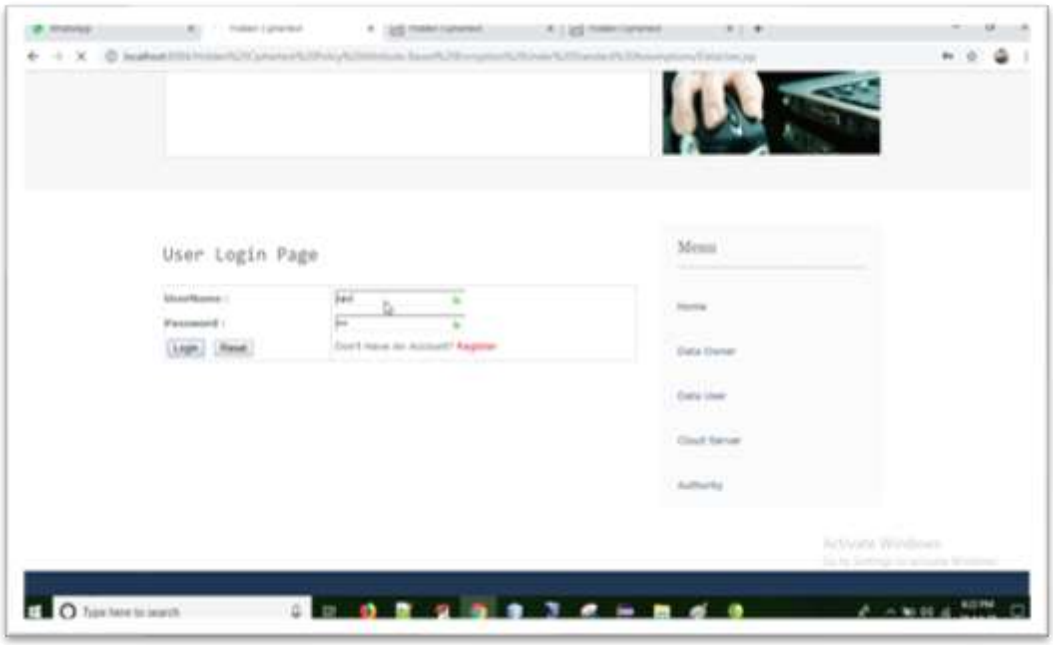

4
suresh
as soon as possible ,, you have to
admit in the hospital
Submit
Back

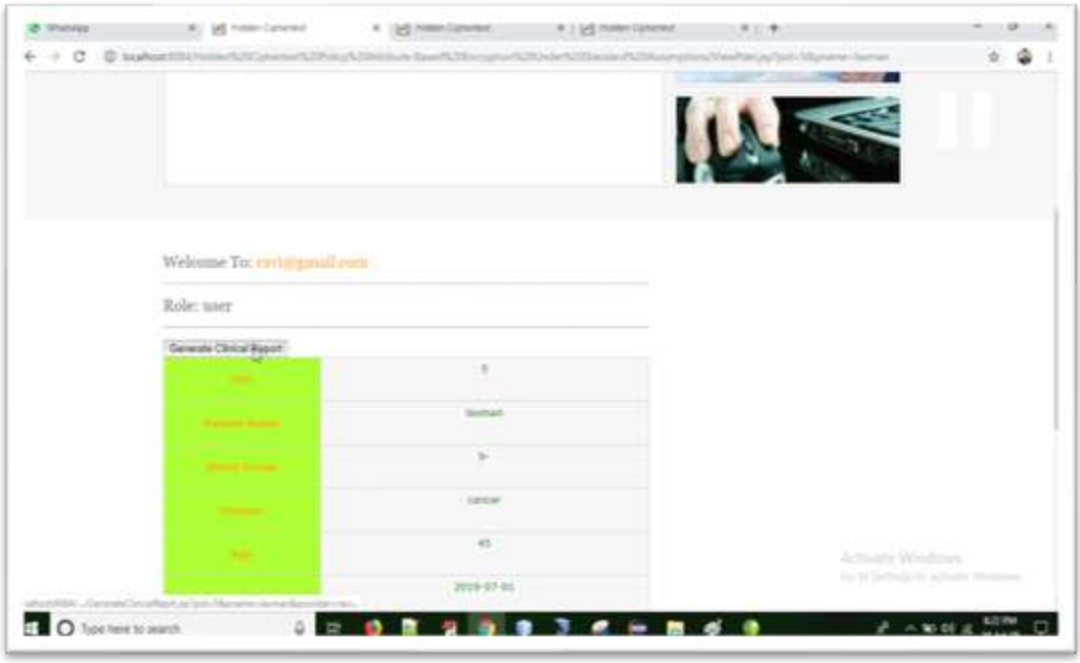

Welcome to our website
Hidden Ciphertext Policy Attribute-Based
Encryption Under Standard Assumptions
View Patient Details
3
suresh
forward to suresh
Back

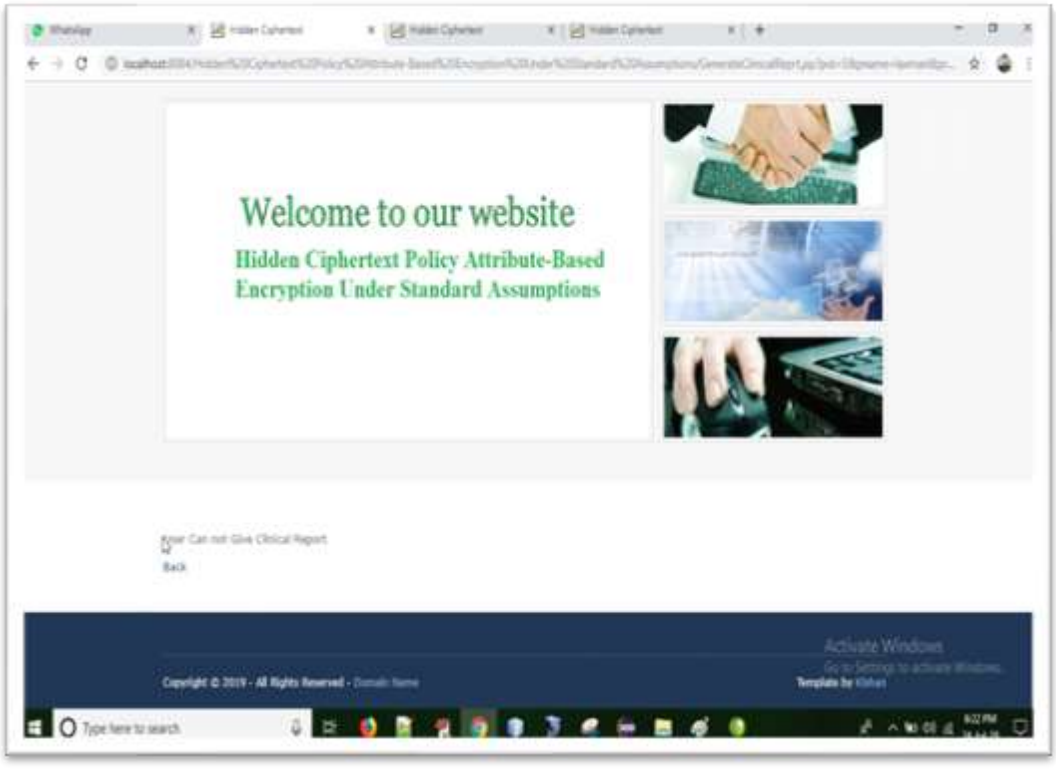

Welcome To:
Role: doctor
as soon as possible , you have to admit in the hospital
Back
Activate Windows


Welcome To:
Encrypt Key:
Select Hospital
submit
Menu
Home
View Profile
Add Patient Details
View Patient Details
Logout
Activate Windows


Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions
Welcome To:
Back
Copyright © 2019 - All Rights Reserved
Activate Windows


User Login Page
Menu
Home
Data Owner
Data User
Cloud Server
Authority
Activate Windows


Welcome To: ravi@gmail.com
Role: user
Generate Clinical Report
1
Stomach
3
cancer
45
2019-07-01


Welcome to our website
Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions
Back

CHAPTER 10

CONCLUSION

In this paper, we introduce a new method called linear secret sharing with multiple values, which can greatly improve the expression of access policy. Moreover, each attribute is divided into two parts, namely the attribute name and its value. Therefore, the most obvious advantage of the proposed scheme is that sensitive attribute values can be hidden. And it can protect users' privacy well in PHR. In the proposed scheme, the size of public parameters is constant and the cost of the decryption is only two pairing operations, which also make it more practical. Eventually, we prove the full security of the proposed scheme in the standard model under static assumptions by using the dual system encryption method. The proposed scheme only achieves partly hiding policy. It is an interesting problem that achieves fully hiding policy with fast encryption, which is left as a future work.

CHAPTER 11

REFERENCES

1. B. Waters, “Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions,” in Advances in Cryptology—CRYPTO (Lecture Notes in Computer Science), vol. 5677, S. Halevi, Eds. Berlin, Germany: Springer, Aug. 2009, pp. 619–636.

2. M. Qutaibah, S. Abdullatif, and C.T. Viet, “A Ciphertext-Policy Attribute based Encryption Scheme With Optimized Ciphertext Size And Fast Decryption,” in Proc. 2017 ACM Asia Conf. Comput, Commun. Secur. (ASIA CCS), Apr. 2017, pp. 230–240.

3. B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in Public Key Cryptography—PKC(Lecture Notes in Computer Science), vol. 6571. Berlin, Germany: Springer, Mar. 2011, pp. 53–70.

4. V.Goyal,O.Pandey,A.Sahai,andB.Waters,“Attribute-based encryption forfine-grainedaccesscontrolofencrypteddata,”inProc.13thACMConf. Comput, Commun. Secur. (CCS), Nov. 2006, pp. 89–98.

5. J. Lai, R.H. Deng, and Y. Li, “Expressive CP-ABE with partially hidden accessstructures,”inProc.7thACMSym.Infor.,Comput,Commun.Secur., May. 2012, pp. 18–19.

6. A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in Advances in Cryptology—EUROCRYPT (Lecture Notes in Computer Science), vol. 3494,R.Cramer,Eds.Berlin,Germany:Springer,May2005,pp.457–473.

7. J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,”in Proc. IEEE Symp. Secur.Privacy(SP),May2007,pp.321– 334.

8. Y. Zhang, D. Zheng, and R.H. Deng, “Security and privacy in smart health: Efficient policy-hiding attribute-based access control,” IEEE Internet Things J., vol. 5, no. 3, pp. 2130–2145, Jun. 2018

9. H. Cui, R.H. Deng, G. Wu, and J. Lai, “An Efficient and Expressive Ciphertext-Policy Attribute-Based Encryption Scheme with Partially Hidden Access Structures,” in Provable Security—PROVSEC (Lecture Notes in Computer Science), vol. 10005, L. Chen, Eds. Berlin, Germany: Springer, Nov. 2016, pp.19–38.

10. C.Y. Umesh, “Ciphertext-policy attribute-based encryption with hiding access structure,” in IEEE Inter.Adv.Comput. Conf. (IACC), Jul 2015, pp. 6–10