CHAPTER 1

CONTENTS OF THE BASE PAPER

1.1 PAPER DETAILS

A Secure Elliptic Curve Cryptography Based Mutual Authentication Protocol for Cloud-assisted TMIS.

Telematics and Informatics, Science Direct, September 2018, indexed in SCOPUS.

Paper Authors: Vinod Kumar, Musheer Ahmad, Adesh Kumari.

URL: https://doi.org/10.1016/j.tele.2018.09.001

1.2 INTRODUCTION

In this paper author is describing concept to provide security to patient data which is store at cloud servers. Now-a-days Telecare Medical Information System (TIMS) using in almost all advance countries to provide treatment to patient remotely. Existing method require doctor or patient to visit each other in order to get or provide treatment. But TIMS not require any manual visits for treatment as hospital will register patient and embed sensor in patient body and that sensor monitor patient health condition and report to patient mobile and patient mobile will report to concern hospital. Hospital people will analyze that data and send require prescription to patient mobile upon abnormal conditions detected. All medical data of patients will be store at cloud servers and this data can be access by multiple hospitals also.

Above describe advantages will not provide security to patient data as the data store at cloud which is not secure. To provide security existing algorithms will encrypt patient data but that data can be attacked and there is no privacy to patient data.

To prevent attack and to provide privacy to patient data author has introduce secure Elliptic Curve Cryptography (ECC). The advantage of this ECC is its key size is small compare to existing technique and it takes less computation time. This technique provides privacy to patient data by encryption and anonymity (hide patient name with *) technique.

ECC secure from attack as each access by doctor or patient or any other third party access will be authenticated with random number given at registration time. If this random no generated key matched then only doctor or patient or cloud peoples allowed to access patient data.

1.3 PROPOSED SYSTEM

Proposed scheme:

We proposed the enhance protocol of Li et al. scheme in the same environment i.e. by providing security and authentication through the implementation of elliptic curve cryptography to meet all the required security properties.

The enhanced framework has many important characteristics, which are demonstrated following as:

- Authentication is established between healthcare center and patient and doctor.
- Patient and healthcare center manage the stability of security by the help of TMIS and send information in TMIS using algorithm
- Further, the presented framework is strong against man-in-the-middle attack, patient anonymity, replay attack, known-key security property, data confidentiality, data non-repudiation, message authentication, impersonation attack, session key security, patient unlinkability and doctor unlinkability.
- We evaluated the proposed protocol with other presented protocols in same environment and found that it is secure and efficient in terms of communication and computation cost.

Preliminaries:

Elliptic Curve Cryptography:

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers.

Def: An elliptical curve can simply illustrate as a set of points over a prime field Z_P defined by the weierstrass equation:

$$Y^2 = (X^3+aX+b) \bmod p, \; where \; (4a^3 + 27b^2) \bmod p \neq 0 \text{ (Non-singular curve)}$$

Arithmetic operation on ECC:

- POINT ADDITION:

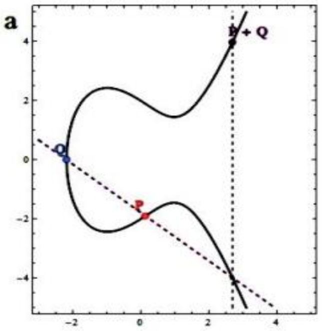


Fig1.a

$P(x_1, y_1)$ & $Q(x_2, y_2)$ be two points on the curve then

$\Rightarrow P + Q = R(x_3, y_3)$

Where $x_3 = \{\lambda 2 - x_1 - x2\} \bmod p$, $y3 = \{\lambda(x_1 - x_3) - y_1\} \bmod p$

$\lambda = \{(y2 - y_1)/(x2 - x_1)\} \bmod p$

- POINT DOUBLING:

If P=Q then P+P=2P

Where $x_3 = \{\lambda^2 - 2\, x_1\} \bmod p$, $y3 = \{\lambda(x_1 - x_3) - y_1\} \bmod p$

$\lambda = (3\, x_1^2 + a)/2\, y_1 \bmod p$

- POINT AT INFINITY:

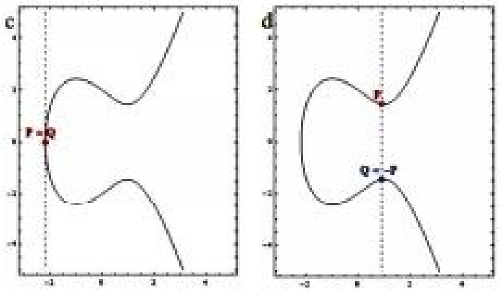


Fig1.b

Which is also known as Ideal point on the curve?

If $x_1 = x_2$ & $y_1 = y_2 = 0$ or $x_1 = x_2$ & $y_1 = - y_2$, then

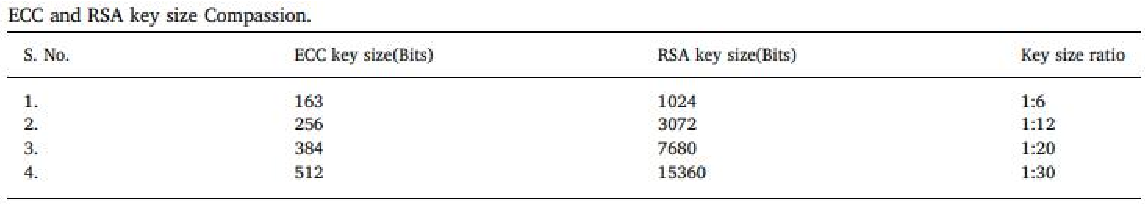
The points is said to intersect at infinity denoted by O.

- POINT NEGATION:
  Point negation is finding such a point, that adding it to itself will result in point at infinity P + (-Q) = O

- ELLIPTIC CURVE POINT MULTIPLICATION is the operation of successively adding a point along an elliptic curve to itself repeatedly. It is used in elliptic curve cryptography (ECC) as a means of producing a one-way function.
  N*P= P+P+P+P+P…. N times, where N is scalar.

Why ECC?
ECC can yield a level of security with a 164-bit key that other systems (RSA) require a 1,024-bit key to achieve. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications.

Table1



ECC and RSA key size Compassion.

S. No.	ECC key size(Bits)	RSA key size(Bits)	Key size ratio
1.	163	1024	1:6
2.	256	3072	1:12
3.	384	7680	1:20
4.	512	15360	1:30

Elliptic Curve Discrete Logarithms problem (ECDLP): For given R S G, find t in Z_q such that S=tR, which is hard.Elliptic Curve Computational Diffie-Hellman problem (ECCDHP): For c d in Zq and g is the base point of G, for given (g, cg, dg), then to compute c d g is hard for the group G.

Algorithm used in Proposed System:
For Message security Encryption and Decryption Process

KEY GENERATION:-
Given a, b: - Coefficients for defining curve , G :- Generator point

Let n: - Order of G such that n*G=O(identity element)

- ✓ Sender selects Private Key-N_a in [1, n-1],Public key - $P_a = N_a$*G.
- ✓ Receiver selects Private key N_b ,Public key P_b= N_b * G.
- ✓ Security key is calculated by sender - K= N_a * P_b ,by receiver - K= N_b * P.

ENCRYPTION ALGORITHM:-

Sender =>(message m)=> receiver

- ✓ Let m has any point M on the elliptic curve let it be $\mathbf{P_m}$
- ✓ Random number k in [1,n-1].
- ✓ The cipher text - $\mathbf{P_c = \{k*G , P_m + k*P_b\}}$

ALGORITHM:-

Plain Text - $\mathbf{P_m = \{P_m + k*P_b - N_b*k*G\}}$

1.3.1 HEALTHCARE CENTER UPLOAD PHASE

In this phase patient register at hospital and hospital will store patient details and gave patient one random number for future authentication. ECC public and private key will be generated on this random number and data will be encrypted. If patient provide valid random no then only he will be authenticated at cloud and allow storing new records and access old records with doctor prescription.

1.3.2 PATIENT DATA UPLOAD PHASE

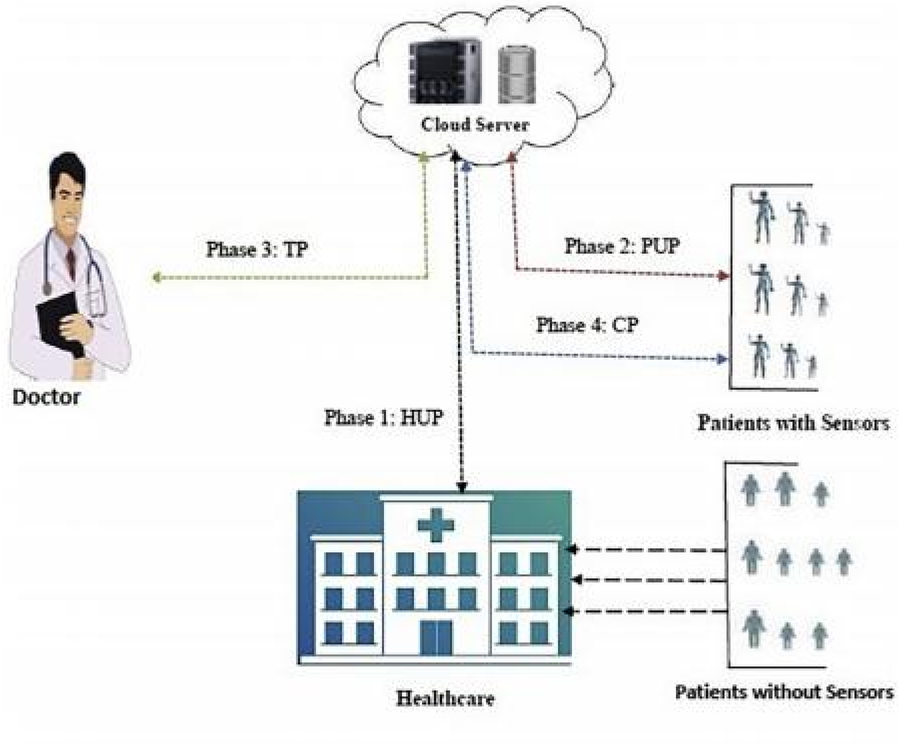
In this phase patient sensor will read patient body data and send to mobile and mobile send to hospital and if patient mobile has valid random no then it will authenticated and send to cloud. Here we don’t have sensors so we will write values in application.

1.3.3 TREATMENT PHASE

In this phase doctor, he authenticate with cloud by using random no given by admin. If valid random no then genuine keys and hash value will be generated and doctor allow to access patient data.

1.3.4 CHECKUP PHASE

In this phase patient authenticate himself with cloud using genuine random no. After authentication patient can view his records and doctor prescription. If random no is not correct then access will be denied.



Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)

Fig2. Architecture of proposed model with different phases

CHAPTER 2

MERITS AND DEMERITS OF THE BASE PAPER

2.1 Existing Solution

Authentication is a significant security prerequisite for secure communications. However, most of identity assisted mutual authentication protocols are requiring of security verification, which is extremely impartment for cryptographic approach. Early three-factor authentication directly employ biometrics as an authentication factor and do not consider the security properties of session key agreement and perfect forward secrecy. used elliptic curve cryptography to achieve the functionalities of mutual authentication and perfect forward secrecy.

2.2 Merits

we have proposed an authentication scheme based on ECC and cloud servers. In the proposed scheme in this paper, we have formally analyzed the security properties of the designed scheme by the most widely accepted and used Automated Validation of Internet Security Protocols and Applications tool. Security and performance analysis show that when compared with other related schemes, the proposed scheme is more powerful, efficient, and secure with respect to various known attacks.

2.3 Demerits

Illegal access to the system information is a potential risk which can cause many problems varies from compromising the user’s privacy to given wrong medical service to a patient which lead to his/her death. Hence, during the last decade, many researchers tried to address these concerns by providing various protocols for access control, authentication and key management in such applications.

CHAPTER 3

SOURCE CODE

3.1 JAVA :

3.1.1 Admin.java

```
package com;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class Admin extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {

    response.setContentType("text/html");

    HttpSession session=request.getSession();

    PrintWriter out = response.getWriter();

    String user=request.getParameter("t1");

    String pass=request.getParameter("t2");

    try{

        if(user.equals("admin") && pass.equals("admin")){

            session.setAttribute("user",user);

RequestDispatcher rd=request.getRequestDispatcher("AdminScreen.jsp?t1=Welcome
"+user);

       rd.forward(request, response);
```

```
                }else{
                        response.sendRedirect("Admin.jsp?t1=Invalid User");
                }
        }catch(Exception e){
                e.printStackTrace();
        }
}
}
```

3.1.2 Database Connection(JDBC):

```
package com;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Calendar;
import java.sql.Statement;
import java.util.ArrayList;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.File;
import org.jfree.ui.RefineryUtilities;
public class DBConnection{
 private static Connection con;
public static Connection getCon()throws Exception {
  try{
 Class.forName("com.mysql.jdbc.Driver");
 con = DriverManager.getConnection("jdbc:mysql://localhost/tims","root","root");
   }catch(Exception e){
```

```
            e.printStackTrace();}
      return con;
}
public static String anonymity(String name){
      int length = name.length()-4;
      if(length < 0)
            name = "eeeeeee"+name;
      length = name.length()-4;
      StringBuilder sb = new StringBuilder();
      sb.append(name.substring(0,length));
      sb.append("****");
      return sb.toString();}
public static String addPhysician(String[] input)throws Exception{
  String msg="fail";
  boolean flag=false;
      int random = 4444;
      con = getCon();
      Statement stmt=con.createStatement();
  ResultSet rs=stmt.executeQuery("select count(*) from adddoctor");
  if(rs.next()){
            random = random + rs.getInt(1);
      }
      random = random + 1;
      stmt=con.createStatement();
  rs=stmt.executeQuery("select username from adddoctor where
username='"+input[0]+"'");
  if(rs.next()){
    flag=true;
    msg = "Username already exist";
  }else{
```

```
            ECC ecc = new ECC();
            String public_key = ecc.loadKeys(random+"",random+"",random+"");
            PreparedStatement stat=con.prepareStatement("insert into adddoctor values(?,?,?,?,?,?,?,?,?)");
            stat.setString(1,input[0]);
            stat.setString(2,input[1]);
            stat.setString(3,input[2]);
            stat.setString(4,input[3]);
            stat.setString(5,input[4].trim());
            stat.setString(6,input[5].trim());
            stat.setString(7,input[6].trim());
            stat.setString(8,random+"");
            stat.setBytes(9,public_key.getBytes());
            int i=stat.executeUpdate();
            if(i > 0){msg = "success,"+random;}
  }
  return msg;
}
public static String createProfile(String[] input)throws Exception{
  String msg="fail";
  int pid = 0;
      int random = 1234;
  con = getCon();
      Statement stmt=con.createStatement();
  ResultSet rs=stmt.executeQuery("select count(*) from patientprofile");
  if(rs.next()){
    pid = pid + rs.getInt(1);
  }
      pid = pid + 1;
      random = random + pid;
```

```
        ECC ecc = new ECC();
        String public_key = ecc.loadKeys(random+"",random+"",random+"");
        java.util.Date d1 = new java.util.Date(input[1].trim());
        java.sql.Date d2 = new java.sql.Date(d1.getTime());
  PreparedStatement stat=con.prepareStatement("insert into patientprofile
values(?,?,?,?,?,?,?,?,?,?,?)");
        stat.setString(1,"PID-"+pid);
        stat.setString(2,input[0]);
  stat.setDate(3,d2);
  stat.setString(4,input[2]);
  stat.setString(5,input[3]);
  stat.setString(6,input[4].trim());
        stat.setString(7,input[5].trim());
        stat.setString(8,input[6].trim());
        stat.setString(9,input[7]);
        stat.setString(10,random+"");
        stat.setBytes(11,public_key.getBytes());
        int i=stat.executeUpdate();
  if(i > 0){ msg = "success,"+"PID-"+pid+","+random;}
  return msg;
}
public static String addPrescription(String[] input)throws Exception{
  String msg="no";
  con = getCon();
        java.util.Date d1 = new java.util.Date();
        java.sql.Timestamp d2 = new java.sql.Timestamp(d1.getTime());
  PreparedStatement stat=con.prepareStatement("insert into prescription values(?,?,?,?)");
  stat.setString(1,input[0]);
  stat.setString(2,input[1]);
  stat.setString(3,input[2]);
```

```
   stat.setTimestamp(4,d2);
         int i=stat.executeUpdate();
   if(i > 0){
  msg = "success";
         }
   return msg;
}
public static String login(String input[])throws Exception{
   String msg="invalid login";
         ECC ecc = new ECC();
         String key = ecc.loadKeys(input[2],input[2],input[2]);
         System.out.println("Generated Random Key : "+key);
   con = getCon();
   Statement stmt=con.createStatement();
   ResultSet rs=stmt.executeQuery("select public_key from adddoctor where
username='"+input[0]+"' and password='"+input[1]+"'");
   if(rs.next()){
      String pk = new String(rs.getBytes(1));
                  if(pk.equals(key)){
                           msg = "success";
                           System.out.println("Keys Matched Successfully");
                  }
   }
   System.out.println(msg);
   return msg;
}
public static String patientLogin(String pid,String random)throws Exception{
   String msg="invalid login";
         ECC ecc = new ECC();
         String key = ecc.loadKeys(random,random,random);
```

```
        System.out.println("Generated Random Key : "+key);
   con = getCon();
   Statement stmt=con.createStatement();
   ResultSet rs=stmt.executeQuery("select public_key from patientprofile where patient_id='"+pid+"'");
   if(rs.next()){
      String pk = new String(rs.getBytes(1));
                if(pk.equals(key)){
                        msg = "success";
                        System.out.println("Keys Matched Successfully");
                }
   }
   return msg;
}
public static void savePrescription(String pid,String record,String prescription,String doctor)throws Exception{
        con = getCon();
        java.util.Date d1 = new java.util.Date();
        java.sql.Timestamp time = new java.sql.Timestamp(d1.getTime());
        PreparedStatement stat=con.prepareStatement("insert into prescription values(?,?,?,?,?)");
        stat.setString(1,pid);
        stat.setString(2,record);
        stat.setString(3,prescription);
        stat.setString(4,doctor);
        stat.setTimestamp(5,time);
        stat.executeUpdate();
}
public static String getPrescription(String pid,String record)throws Exception{
   String msg="none";
```

```
            con = getCon();
    Statement stmt=con.createStatement();
    ResultSet rs=stmt.executeQuery("select prescription,doctor_name,date_time from
prescription where patient_id='"+pid+"' and record_no='"+record+"'");
    if(rs.next()){
        msg = rs.getString(1)+","+rs.getString(2)+","+rs.getTimestamp(3).toString();
            }
    return msg;
}
public static String getRandom(String pid)throws Exception{
    String msg="invalid login";
            con = getCon();
    Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select random_no from patientprofile where
patient_id='"+pid+"'");
    if(rs.next()){
        msg = rs.getString(1);
            }
    return msg;
}}
```

3.1.3 Elliptic Curve.java

```
package com;
import java.math.BigInteger;
import java.io.Serializable;
public class EllipticCurve implements Serializable {
            // The parameters of an EC.
            private BigInteger p;
            private BigInteger a;
```

```
        private BigInteger b;
public EllipticCurve(BigInteger prime, BigInteger myA, BigInteger myB) {
        p = prime;
        a = myA;
        b = myB;
}
// Copy constructor.
public EllipticCurve(EllipticCurve copy) {
        p = new BigInteger(copy.p.toString());
        a = new BigInteger(copy.a.toString());
        b = new BigInteger(copy.b.toString());
}
// All three components must be equal for the curves to be the same.
public boolean equals(EllipticCurve other) {
        return p.equals(other.p) && a.equals(other.a) && b.equals(other.b);
}
public BigInteger getP() {
        return p;
}
public BigInteger getA() {
        return a;
}
}
```

3.1.4 ECC.java

```
package com;
import java.math.BigInteger;
import java.util.Random;
import java.io.Serializable;
import java.io.FileInputStream;
import java.util.ArrayList;
public class ECC implements Serializable{
        // Parts of one ECC system.
        EllipticCurve curve;
        Point generator;
        Point publicKey;
        BigInteger privateKey;
        public Point[] encrypt(Point plain) {
        // First we must pick a random k, in range.
        int bits = curve.getP().bitLength();
        BigInteger k = new BigInteger(bits, new Random());
        System.out.println("Picked "+k+" as k for encrypting.");
        // Our output is an ordered pair, (k*generator, plain + k*publickey)
        Point[] ans = new Point[2];
        ans[0] = generator.multiply(k);
        ans[1] = plain.add(publicKey.multiply(k));
        return ans;
}
// Decryption - notice the similarity to El Gamal!!!
public Point decrypt(Point[] cipher) {
        // This is what we subtract out.
        Point sub = cipher[0].multiply(privateKey);
        // Subtract out and return.
        return cipher[1].subtract(sub);
}
public String toString() {
        return "Gen: "+generator+"\n"+"pri: "+privateKey+"\n"+"pub: "+publicKey;
}
```

```
public String loadKeys(String id,String random,String pk){

        // Just use the book's curve and test.

        BigInteger bi = new
BigInteger("66034722589109753904093868754190606709127111586509026735934726
81492443618519247014208029797276135537622513121022070188456656102492281 1
01483976551651052349400660865106891558505875353446226922766370854915854 1
92720967802181687778812897356989588519483299949526013215505901680565031 4
99772113902446208729069");

        curve = new EllipticCurve(bi,bi,bi);

        BigInteger x = new BigInteger(id);

        BigInteger y = new BigInteger(random);

        BigInteger nA = new BigInteger(pk);

        generator = new Point(curve, x, y);

        privateKey = nA;

        publicKey = generator.multiply(privateKey);

        return publicKey.toString();

}

public ECC getKeys(){

        return this;

}

/*public static void main(String args[])throws Exception{

        //BigInteger prime = BigInteger.probablePrime(1000,new Random());

        //System.out.println(prime+"\n\n");

        FileInputStream fin = new FileInputStream("tt.txt");

        byte b[] = new byte[fin.available()];

        fin.read(b,0,b.length);

        fin.close();

        ECC ecc = new ECC();

        String bi = "1235";//"14893003337626352152463254152616458181260144281";

        ecc.loadKeys(bi,bi,bi);
```

```
	String data = new String(b);
	String arr[] = data.split("\n");
	ArrayList<Point[]> list = new ArrayList<Point[]>();
	for(int i=0;i<arr.length;i++) {
		Point plain = new Point(ecc.curve, new BigInteger(arr[i].getBytes()), new BigInteger("kk".getBytes()));
		Point[] cipher = ecc.getKeys().encrypt(plain);
		list.add(cipher);
		System.out.println(i);


	}
	StringBuilder sb = new StringBuilder();
	for(int i=0;i<list.size();i++){
		Point decrypt = ecc.getKeys().decrypt(list.get(i));
		byte b1[] = decrypt.getX().toByteArray();
		byte b2[] = decrypt.getY().toByteArray();
		System.out.println("value = "+new String(b1)+" "+new String(b2));
	}
	System.out.println(ecc.toString());
}*/
}
```

3.1.5 Healthcare upload.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

public class HealthcareUpload extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String pname=request.getParameter("t1");

        String bdate=request.getParameter("t2");

        String age=request.getParameter("t3");

        String gender=request.getParameter("t4");

        String contact=request.getParameter("t5");

        String address=request.getParameter("t6");

        String problem=request.getParameter("t7");

        String desc=request.getParameter("t8");

        try{

                String input[]={pname,bdate,age,gender,contact,address,problem,desc};

                String msg=DBConnection.createProfile(input);

                if(!msg.equals("fail")){

                        String arr[] = msg.split(",");

                        RequestDispatcher
rd=request.getRequestDispatcher("AdminScreen.jsp?t1=Your Profile
created.<br/>Patient ID = "+arr[1]+"<br/>Generated Random No : "+arr[2]);


                rd.forward(request, response);

                }

                else{

                        response.sendRedirect("AdminScreen.jsp?t1=Error in creating
pfofile");

                }
```

```
          }
          catch(Exception e)
          {e.printStackTrace();}
      }
}
```

3.1.6 Patient.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class Patient extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)    throws
ServletException, IOException {
          response.setContentType("text/html");
          HttpSession session=request.getSession();
          PrintWriter out = response.getWriter();

          String pid=request.getParameter("t1");
          String random=request.getParameter("t2");
          try{
                    String msg=DBConnection.patientLogin(pid,random);
                    boolean flag=false;
                    if(msg.equals("success")){
                              session.setAttribute("user",pid);
                              session.setAttribute("random",random);
                              RequestDispatcher
rd=request.getRequestDispatcher("PatientScreen.jsp?t1=Welcome "+pid);
                              rd.forward(request, response);
                    }else{
                              response.sendRedirect("Patient.jsp?t1=Invalid Patient ID");
                    }



          }catch(Exception e){
                    e.printStackTrace();
          }
          }
}
```

3.1.7 PatientDataUpload.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.util.ArrayList;
import java.io.File;
import java.io.ObjectOutputStream;
import java.io.ObjectInputStream;
import java.io.FileOutputStream;
import java.io.FileInputStream;

import java.math.BigInteger;

public class PatientDataUpload extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String pid=request.getParameter("t1");

        String random=request.getParameter("t2");

        String bp=request.getParameter("t3");

        String heart=request.getParameter("t4");

        String problem=request.getParameter("t5");

        try{

                ArrayList<Point[]> list = null;

                String path = getServletContext().getRealPath("/")+"WEB-
INF/patients/"+pid+".txt";

                File file = new File(path);



        if(file.exists()) {

                        ObjectInputStream oin = new ObjectInputStream(new
        FileInputStream(file));

                        Object obj = (Object)oin.readObject();

                        list = (ArrayList<Point[]>)obj;

                        oin.close();

                } else {

                        list = new ArrayList<Point[]>();

                }

                String data = bp+","+heart+","+problem;

                ECC ecc = new ECC();

                ecc.loadKeys(random,random,random);
```

```
                    Point plain = new Point(ecc.curve, new BigInteger(data.getBytes()), new
BigInteger("kk".getBytes()));

                    Point[] cipher = ecc.getKeys().encrypt(plain);

                    list.add(cipher);

                    ObjectOutputStream oout = new ObjectOutputStream(new
FileOutputStream(file));

                    oout.writeObject(list);

                    oout.close();

                    RequestDispatcher
rd=request.getRequestDispatcher("PatientDataUpload.jsp?t1=Your data uploaded to
cloud inside WEB-INF/patients folder");

                    rd.forward(request, response);

          }catch(Exception e){

                    e.printStackTrace();

          }

  }}
```

3.1.8 Physician.java

```
package com;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;


public class Physician extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {
        response.setContentType("text/html");
        HttpSession session=request.getSession();
        PrintWriter out = response.getWriter();
        String user=request.getParameter("t1");
        String pass=request.getParameter("t2");
        String random=request.getParameter("t3");
        try{
                String input[]={user,pass,random};
                String msg=DBConnection.login(input);
                boolean flag=false;
                if(msg.equals("success")){
                        session.setAttribute("user",user);
                        flag = true;
                        RequestDispatcher
rd=request.getRequestDispatcher("DoctorScreen.jsp?t1=Welcome "+user);
                        rd.forward(request, response);
                }else{
                        response.sendRedirect("Doctor.jsp?t1=Invalid User");
                }
        }catch(Exception e){
                e.printStackTrace();
        }
   }
}
```

3.1.9 Point.java

```
package com;
import java.math.BigInteger;
import java.io.Serializable;
public class Point implements Serializable{
        // Store the x, y and curve.
        BigInteger x;
        BigInteger y;
        EllipticCurve curve;
        String doctor,treatment;

public void setDoctor(String doctor){
        this.doctor = doctor;
}
public String getDoctor(){
        return doctor;
}

public void setTreatment(String treatment){
        this.treatment = treatment;
}
public String getTreatment(){
        return treatment;
}
// Precondition: (myX, myY) must lie on the curve c. I don't check that here!!!
public Point(EllipticCurve c, BigInteger myX, BigInteger myY) {
        x = myX;
        y = myY;
        curve = c;
}
// Copy constructor.
public Point(Point copy) {
        x = new BigInteger(copy.x.toString());
        y = new BigInteger(copy.y.toString());
        curve = new EllipticCurve(copy.curve);
```

```
}

// Returns 0. Not sure if this is the proper way to store the "origin".

public Point(EllipticCurve c) {

        curve = c;

        x = BigInteger.ZERO;

        y = BigInteger.ZERO;

}

// All components must be equal...

public boolean equals(Point other) {

        return x.equals(other.x) && y.equals(other.y) && curve.equals(other.curve);

}

// Returns true iff other is this point's reflection over the line y = p/2 (real division)

public boolean mirror(Point other) {

        return x.equals(other.x) && curve.equals(other.curve) &&
y.equals(other.curve.getP().subtract(other.y));

}

// Returns the negative of this point, which is its mirror.

public Point negate() {

        BigInteger newY = curve.getP().subtract(y);

        return new Point(curve, x, newY);

}

// Adds this to other and returns the answer, using the formulas in Stallings (5th edition)

public Point add(Point other) {

        // Can't add points on different curves.

        if (!curve.equals(other.curve))

                return null;

        if (this.equals(other)) {

                // We need these to calculate lambda.

                BigInteger three = new BigInteger("3");

                BigInteger two = new BigInteger("2");
                BigInteger temp = new BigInteger(x.toString());

                // Splitting up the calculation of lambda into all of these steps...

                BigInteger lambda = temp.modPow(two, curve.getP());

                lambda = three.multiply(lambda);

                lambda = lambda.add(curve.getA());

                BigInteger den = two.multiply(y);

                lambda = lambda.multiply(den.modInverse(curve.getP()));

                // Once we have lambda, just plug into these equations.

                BigInteger newX =
lambda.multiply(lambda).subtract(x).subtract(x).mod(curve.getP());

                BigInteger newY =
(lambda.multiply(x.subtract(newX))).subtract(y).mod(curve.getP());

                return new Point(curve, newX, newY);

        }

        // Returns the origin...not sure if my origin is correct.

        else if (this.mirror(other)) {

                return new Point(curve);

        }

        // Standard case.

        else {

                // We need these to calculate lambda.

                BigInteger three = new BigInteger("3");

                BigInteger two = new BigInteger("2");

                BigInteger temp = new BigInteger(x.toString());

                // Lambda's a bit easier here...

                BigInteger lambda = other.y.subtract(y);

                BigInteger den = other.x.subtract(x);

                lambda = lambda.multiply(den.modInverse(curve.getP()));

                // This calculation is roughly the same as above.
```

```
                BigInteger newX =
lambda.multiply(lambda).subtract(x).subtract(other.x).mod(curve.getP());
                BigInteger newY =
(lambda.multiply(x.subtract(newX))).subtract(y).mod(curve.getP());
                return new Point(curve, newX, newY);
        }
}
// Subtraction is just adding the negative.
public Point subtract(Point other) {
        other = other.negate();
        return this.add(other);
}
// Uses "fast multiplication" to multiply this point by factor.
public Point multiply(BigInteger factor) {
        BigInteger two = new BigInteger("2");
        // Base cases.
        if (factor.equals(BigInteger.ONE))
                return new Point(this);
        if (factor.equals(two))
                return this.add(this);
        // Even case where we can calculate half of our answer and multiply by 2!
        if (factor.mod(two).equals(BigInteger.ZERO)) {
                Point sqrt = multiply(factor.divide(two));
                return sqrt.add(sqrt);
        }
        // No speed up here, but this recursive call will lead to one.
        else {
                factor = factor.subtract(BigInteger.ONE);

        return this.add(multiply(factor));
```

```
	}
}
public String toString() {
	return x+","+y;
}
public BigInteger getX(){
	return x;
}
public BigInteger getY(){
	return y;
}
}
```

3.1.10 ECDSA.java

```
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Formatter;
public class MessageSV {
	/*
	 * function: messageSign - sign the message using the private key
	 * returns: Array kG containing (r,s) - signature
	 */
	public static BigInteger[] messageSign(String msg, BigInteger n, BigInteger[] G,
BigInteger a, BigInteger pvkd) throws NoSuchAlgorithmException {
		System.out.println("Signing the message ....");
		BigInteger k, kInv, r, e, s;
		BigInteger[] kG;
		do {
```

```
            do {
                k = BigInteger.valueOf(1);
                kG = ECOperations.pointMultiply(G, n, a, k);
                r = kG[0].mod(n);
            } while (r.compareTo(BigInteger.ZERO) == 0);

            kInv = k.modInverse(n);
            e = new BigInteger(SHAsum(msg.getBytes()), 16);
            s = (kInv.multiply(e.add(pvkd.multiply(r)))).mod(n);
        } while (s.compareTo(BigInteger.ZERO) == 0);

        kG[0] = r;
        kG[1] = s;
        System.out.println("Message SIGNED");
        return kG;
    }
/*
    * function: messageVerify - sign the message using the private key
    * returns: boolean  - verification Status
    */
public static boolean messageVerify(String msg, BigInteger[] sign, BigInteger n,
BigInteger[] G, BigInteger a, BigInteger[] pbkQ) throws NoSuchAlgorithmException {
        System.out.println("Verifying Message ......");
        BigInteger r = sign[0];
        BigInteger s = sign[1];
        if (r.compareTo(n) >= 0) {
            System.out.println("Message NOT VERIFIED");
            return false;
        }
        if (s.compareTo(n) >= 0) {
```

```
                        System.out.println("Message NOT VERIFIED");
                        return false;
                }
                BigInteger e = new BigInteger(SHAsum(msg.getBytes()), 16);
                BigInteger w = s.modInverse(n);
                BigInteger u1 = (e.multiply(w)).mod(n);
                BigInteger u2 = (r.multiply(w)).mod(n);
BigInteger[] X = ECOperations.pointAddition(ECOperations.pointMultiply(G, n, a, u1),
ECOperations.pointMultiply(pbkQ, n, a, u2), n);
            BigInteger v = X[0].mod(n);
               if (v.compareTo(r) == 0) {
                        System.out.println("Message VERIFIED");
                        return true;
                }
                System.out.println("Message NOT VERIFIED");
                return false;
        }
public static String SHAsum(byte[] convertme) throws NoSuchAlgorithmException {
            MessageDigest md = MessageDigest.getInstance("SHA-1");
            return byteArray2Hex(md.digest(convertme));
        }
             private static String byteArray2Hex(final byte[] hash) {
            Formatter formatter = new Formatter();
            for (byte b : hash) {
               formatter.format("%02x", b);
            }
            return formatter.toString();
        }
}
```

3.1.11 Prescription.java

```
package com;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.util.ArrayList;

import java.io.File;

import java.io.ObjectOutputStream;

import java.io.ObjectInputStream;

import java.io.FileOutputStream;

import java.io.FileInputStream;

import java.math.BigInteger;

public class Prescription extends HttpServlet {

public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String record=request.getParameter("t1").trim();

        String pid=request.getParameter("t2");

        String prescription=request.getParameter("t3");

        try{

                HttpSession session=request.getSession();

                String doctor = session.getAttribute("user").toString().trim();

                String random = DBConnection.getRandom(pid);

                DBConnection.savePrescription(pid,record,prescription,doctor);
```

```
                    RequestDispatcher
rd=request.getRequestDispatcher("Treatment.jsp?t1=Prescription uploaded to cloud
inside WEB-INF/patients folder");
                    rd.forward(request, response);
          }catch(Exception e){
                    e.printStackTrace();
          }
   }
}
```

3.1.12 RegisterDoctor.java

```
package com;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;


public class RegisterDoctor extends HttpServlet {
public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {
          response.setContentType("text/html");
          HttpSession session=request.getSession();
          PrintWriter out = response.getWriter();
          String user=request.getParameter("t1");
          String pass=request.getParameter("t2");
          String type=request.getParameter("t3");
```

```
        String contact=request.getParameter("t4");
        String address=request.getParameter("t5");
        String qualification=request.getParameter("t6");
        String speciality=request.getParameter("t7");

        try{
                String input[]={user,pass,type,contact,address,qualification,speciality};
                String msg=DBConnection.addPhysician(input);
                if(!msg.equals("fail")){
                        String arr[] = msg.split(",");
                        session.setAttribute("user",user);
                        RequestDispatcher
rd=request.getRequestDispatcher("AdminScreen.jsp?t1=Physician details added
successfully.<br/>Generated Random No : "+arr[1]);
                        rd.forward(request, response);
                }
                else{
                        response.sendRedirect("AdminScreen.jsp?t1="+msg);
                }
        }catch(Exception e){
                e.printStackTrace();
                }
        }}
```

3.2 GUI - JAVA SERVER PAGE:

3.2.1 Admin.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script language="javascript">
        function validate(formObj)
        {
        if(formObj.t1.value.length==0)
        {
        alert("Please Enter Username");
        formObj.t1.focus();
        return false;
        }
        if(formObj.t2.value.length==0)
        {
        alert("Please Enter Password");
        formObj.t2.focus();
        return false;
        }

        formObj.actionUpdateData.value="update";
        return true;
        }
        </script>
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
```

```
        <h1><span><center>A secure elliptic curve cryptography based mutual
authentication

protocol </center><center>for cloud-assisted TMIS</center>
</span><small></small></h1>

      </div>

    </div>

    <div class="content">

      <div class="content_bg">

        <div class="menu_nav">

          <ul>

            <li class="active"><a href="index.jsp">Home</a></li>

            <li><a href="Admin.jsp">Administrator</a></li>

            <li><a href="Doctor.jsp">Doctor Login</a></li>

            <li><a href="Patient.jsp">Patient Login</a></li>

          </ul>

        </div>

        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>

                                                <center>

<form name="f1" method="post" action="Admin" onsubmit="return
validate(this);"><br/>

   <h2><b>Admin Login Screen</b></h2>

     <%

            String res = request.getParameter("t1");

            if(res != null){

                        out.println("<center><font face=verdana
color=red>"+res+"</center></font>");

            }%>



                                                                        <table align="center" width="40" >

                                     <tr><td><b>Username</b></td><td><input type="text"
name="t1" style="font-family: Comic Sans MS" size=20/></td></tr>
```

```
                        <tr><td><b>Password</b></td><td><input type="password" name="t2" 
style="font-family: Comic Sans MS" size=20/></td></tr>

<tr><td></td><td><input type="submit" value="Login"></td>

                        </table>

                                                </div>

                                        </div>

                        </body>

</html>
```

3.2.2 AdminScreen.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" 
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div class="main">

  <div class="main_resize">

    <div class="header">

      <div class="logo">

       <h1><span><center>A secure elliptic curve cryptography based mutual 
authentication

protocol </center><center>for cloud-assisted TMIS</center>
</span><small></small></h1>

      </div>

    </div>

    <div class="content">
    <div class="content_bg">

      <div class="menu_nav">

        <ul>

          <li class="active"><a href="RegisterPatients.jsp">Healthcare Patient Upload
Phase</a></li>

          <li><a href="RegisterDoctor.jsp">Register Doctor</a></li>

                            <li><a href="ViewPatients.jsp">View Patient Details</a></li>

                            <li><a href="ViewDoctors.jsp">View Doctor Details</a></li>



          <li><a href="Logout.jsp">Logout</a></li>

        </ul>

      </div>

      <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>

                  <%

          String res = request.getParameter("t1");

          if(res != null){

                  out.println("<center><font face=verdana
color=red>"+res+"</center></font>");

          }%>

    <p align="justify"><font size="3" style="font-family: Comic Sans MS">Abstract-
With the fast progress of network communication, its technologies and the developing
popularityof telecare medical information system (TMIS), doctors provide treatment to
patients via Internetwithout visiting hospitals. By using mobile device, wireless body area
network and cloud basedarchitecture, the patients can gather their physiological
information and upload to cloud via theirmobile devices. The authenticated doctor
provides online treatment to patient at anytime and anywhere.</p>

 <p align="justify"><font size="3" style="font-family: Comic Sans MS">Moreover,
TMIS maintains security and privacy of the patients in information communicationand
authenticated to all the participants before assessing this system. Recently Liet al. (2018)
presented a cloud-assisted authentication and privacy preservation scheme for
TMIS.They believed that their scheme secure against all well-known privacy and security
attributes. Inthe proposed work, we reviewed Li et al. authentication protocol and found
that it has varioussecurity flaws like as message authentication fails in healthcare center
upload phase, session key
```

```
is not possible in healthcare center upload phase, impersonation attack in patient data
upload phase, patient anonymity and patient unlinkability.</p>

</body>

</body>

</html>
```

3.2.3 CheckupPhase.jsp

```
<%@page import="java.util.ArrayList"%>

<%@page import="java.io.ObjectOutputStream"%>

<%@page import="java.io.ObjectInputStream"%>

<%@page import="java.io.File"%>

<%@page import="java.io.FileInputStream"%>

<%@page import="com.ECC"%>

<%@page import="com.Point"%>

<%@page import="com.DBConnection"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div class="main">

  <div class="main_resize">

    <div class="header">

      <div class="logo">

       <h1><span><center>A secure elliptic curve cryptography based mutual
authentication

protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
```

```
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
         <ul>
            <li class="active"><a href="PatientDataUpload.jsp">Patient Data Upload
Phase</a></li>
            <li><a href="CheckupPhase.jsp">Checkup Phase</a></li>
            <li><a href="Logout.jsp">Logout</a></li>
          </ul>
        </div>
        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                                        <center>
  <h2><b>View Checkup Details Screen</b></h2>
  <%
            String res = request.getParameter("t1");
            if(res != null){
                        out.println("<center><font face=verdana
color=red>"+res+"</center></font>");
            }%>
                        <table border="1" align="center" width="100%">
                        <tr><th><font size="3" color="black">Patient ID</th><th><font
size="3" color="black">Random No</th><th><font size="3" color="black">BP</th>
                        <th><font size="3" color="black">Heart Rate</th><th><font
size="3" color="black">Problem</th>
                        <th><font size="3" color="black">Doctor</th><th><font size="3"
color="black">Prescription</th>
                        <th><font size="3" color="black">Date & Time</th>
                        <tr>
            <%
```

```
	String pid = session.getAttribute("user").toString();

	String random = session.getAttribute("random").toString();

	String path = getServletContext().getRealPath("/")+"WEB-INF/patients/"+pid+".txt";

	File file = new File(path);

	if(file.exists()) {

		ECC ecc = new ECC();

		ecc.loadKeys(random,random,random);

		ObjectInputStream oin = new ObjectInputStream(new FileInputStream(file));

		Object obj = (Object)oin.readObject();

		ArrayList<Point[]> list = (ArrayList<Point[]>)obj;

		oin.close();

		for(int i=0;i<list.size();i++){

			Point decrypt = ecc.getKeys().decrypt(list.get(i));

			byte b1[] = decrypt.getX().toByteArray();

			byte b2[] = decrypt.getY().toByteArray();

			String arr[] = new String(b1).split(",");

			String str = DBConnection.getPrescription(pid,i+"");

		%>

	<tr><td><font size="3" color="black"><%=pid%></td>

	<td><font size="3" color="black"><%=random%></td>

	<td><font size="3" color="black"><%=arr[0]%></td>

	<td><font size="3" color="black"><%=arr[1]%></td>

	<td><font size="3" color="black"><%=arr[2]%></td>

	<%if(!str.equals("none")){

			String temp[] = str.split(",");

			%>

	<td><font size="3" color="black"><%=temp[1]%></td>

	<td><font size="3" color="black"><%=temp[0]%></td>
```

```
        <td><font size="3" color="black"><%=temp[2]%></td>
        <%}}}%>
        </tr>
        </table>
        </body>
</html>
```

3.2.4 Doctor.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script language="javascript">
        function validate(formObj)
        {
        if(formObj.t1.value.length==0)
        {
        alert("Please Enter Username");
        formObj.t1.focus();
        return false;
        }
        if(formObj.t2.value.length==0)
        {
        alert("Please Enter Password");
        formObj.t2.focus();
        return false;
        }
        if(formObj.t3.value.length==0)
```

```
      {
      alert("Please Enter Random Number");
      formObj.t3.focus();
      return false;
      }
      formObj.actionUpdateData.value="update";
      return true;
      }
      </script>
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
       <h1><span><center>A secure elliptic curve cryptography based mutual
authentication protocol </center><center>for cloud-assisted</center>
</span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
          <ul>
            <li class="active"><a href="index.jsp">Home</a></li>
            <li><a href="Admin.jsp">Administrator</a></li>
            <li><a href="Doctor.jsp">Doctor Login</a></li>
            <li><a href="Patient.jsp">Patient Login</a></li>
          </ul>
        </div>
       <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>

                                    <center>

<form name="f1" method="post" action="Physician" onsubmit="return
validate(this);"><br/>

  <h2><b>Doctor Login Screen</b></h2>

   <%

         String res = request.getParameter("t1");

         if(res != null){

                  out.println("<center><font face=verdana
color=red>"+res+"</center></font>");

         }%>



                                                      <table align="center" width="40" >

                             <tr><td><b>Username</b></td><td><input type="text"
name="t1" style="font-family: Comic Sans MS" size=20/></td></tr>



                     <tr><td><b>Password</b></td><td><input type="password" name="t2"
style="font-family: Comic Sans MS" size=20/></td></tr>



                  <tr><td><b>Random No</b></td><td><input type="text" name="t3"
style="font-family: Comic Sans MS" size=20/></td></tr>



                            <tr><td></td><td><input type="submit" value="Login"></td>

                            </table>

                                    </div>

                            </div>

         </body>

</html>
```

3.2.5 DoctorScreen.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
        <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
          <ul>
            <li class="active"><a href="Treatment.jsp">Treatment Phase</a></li>
                                    <li><a href="Logout.jsp">Logout</a></li>
          </ul>
        </div>
        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                        <%
            String res = request.getParameter("t1");
            if(res != null){
                        out.println("<center><font face=verdana color=red>"+res+"</center></font>");

            }%>

         <p align="justify"><font size="3" style="font-family: Comic Sans MS">Abstract-Distributed m-healthcare cloud computing system significantly facilitates efficient patient treatment for medical consultation by sharing personal health information among healthcare providers. However, it brings about the challenge of keeping both the data confidentiality and patients’ identity privacy simultaneously. Many existing access control and anonymous authentication schemes cannot be straightforwardly exploited. To solve the problem, in this paper, a novel authorized accessible privacy model (AAPM) is established. Patients can authorize physicians by setting an access tree supporting flexible threshold predicates. Then, based on it, by devising a new technique of attribute-based designated verifier signature, a patient self-controllable multi-level privacy-preserving cooperative authentication scheme (PSMPA) realizing three levels of security and privacy requirement in distributed m-healthcare cloud computing system is proposed. The directly authorized physicians, the indirectly authorized physicians and the unauthorized persons in medical consultation can respectively decipher the personal health information and/or verify patients’ identities by satisfying the access tree with their own attribute sets. Finally, the formal security proof and simulation results illustrate our scheme can resist various kinds of attacks and far outperforms the previous ones in terms of computational, communication and storage overhead.</p>

      </body>

</html>
```

3.2.6 Index.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div class="main">

  <div class="main_resize">

    <div class="header">

      <div class="logo">
```

```
        <h1><span><center>A secure elliptic curve cryptography based mutual
authentication

protocol </center><center>for cloud-assisted</center> </span><small></small></h1>

      </div>

    </div>

    <div class="content">

      <div class="content_bg">

        <div class="menu_nav">

          <ul>

            <li class="active"><a href="index.jsp">Home</a></li>

            <li><a href="Admin.jsp">Administrator</a></li>

            <li><a href="Doctor.jsp">Doctor Login</a></li>

            <li><a href="Patient.jsp">Patient Login</a></li>

          </ul>

        </div>

        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>

       <p align="justify"><font size="3" style="font-family: Comic Sans MS">Abstract-
With the fast progress of network communication, its technologies and the developing
popularity of telecare medical information system (TMIS), doctors provide treatment to
patients via Internet without visiting hospitals. By using mobile device, wireless body
area network and cloud based architecture, the patients can gather their physiological
information and upload to cloud via their mobile devices. The authenticated doctor
provides online treatment to patient at anytime and anywhere.</p>

 <p align="justify"><font size="3" style="font-family: Comic Sans MS">Moreover,
TMIS maintains security and privacy of the patients in information communication and
authenticated to all the participants before assessing this system. Recently Li et al. (2018)
presented a cloud-assisted authentication and privacy preservation scheme for TMIS.
They believed that their scheme secure against all well-known privacy and security
attributes. In the proposed work, we reviewed Li et al. authentication protocol and found
that it has various security flaws like as message authentication fails in healthcare center
upload phase, session key is not possible in healthcare center upload phase,
impersonation attack in patient data upload phase, patient anonymity and patient
unlinkability.</p>

  </body>

</html>
```

3.2.7 Logout.jsp

```
<%
session.invalidate();
%>
<jsp:forward page="index.jsp" />
```

3.2.8 Patient.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script language="javascript">
        function validate(formObj)
        {
        if(formObj.t1.value.length==0)
        {
        alert("Please Enter Username");
        formObj.t1.focus();
        return false;
        }
        if(formObj.t2.value.length==0)
        {
        alert("Please Enter Random No");
        formObj.t2.focus();
        return false;
        }

        formObj.actionUpdateData.value="update";
```

```
            return true;
            }
            </script>
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
        <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
            <ul>
            <li class="active"><a href="index.jsp">Home</a></li>
            <li><a href="Admin.jsp">Administrator</a></li>
            <li><a href="Doctor.jsp">Doctor Login</a></li>
            <li><a href="Patient.jsp">Patient Login</a></li>
          </ul>
        </div>
        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                                                <center>
<form name="f1" method="post" action="Patient" onsubmit="return
validate(this);"><br/>
   <h2><b>Patient Login Screen</b></h2>
```

```
        <%
        String res = request.getParameter("t1");
        if(res != null){
                out.println("<center><font face=verdana
color=red>"+res+"</center></font>");
        }%>


                                        <table align="center" width="40" >
                        <tr><td><b>Patient&nbsp;ID</b></td><td><input type="text"
name="t1" style="font-family: Comic Sans MS" size=20/></td></tr>


                    <tr><td><b>Random&nbsp;No</b></td><td><input type="text"
name="t2" style="font-family: Comic Sans MS" size=20/></td></tr>
                        <tr><td></td><td><input type="submit" value="Login"></td>
                        </table>
                                </div>
                                </div>
                                </body>
</html>
```

3.2.9 PatientDataUpload.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script language="javascript">
        function validate(formObj)
        {
```

```
        if(formObj.t3.value.length==0)
        {
        alert("Please Enter current blood pressure");
        formObj.t3.focus();
        return false;
        }
        if(formObj.t4.value.length==0)
        {
        alert("Please Enter Current Heart Rate");
        formObj.t4.focus();
        return false;
        }
        if(formObj.t5.value.length==0)
        {
        alert("Please Enter Current Problem");
        formObj.t5.focus();
        return false;
        }
        formObj.actionUpdateData.value="update";
        return true;
        }
        </script>
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
        <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
```

```
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
            <ul>
            <li class="active"><a href="PatientDataUpload.jsp">Patient Data Upload
Phase</a></li>
            <li><a href="CheckupPhase.jsp">Checkup Phase</a></li>
            <li><a href="Logout.jsp">Logout</a></li>
          </ul>
        </div>
        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                                                <center>
<form name="f1" method="post" action="PatientDataUpload" onsubmit="return
validate(this);"><br/>
   <h2><b>Patient Login Screen</b></h2>
 <%
            String res = request.getParameter("t1");
            if(res != null){
                        out.println("<center><font face=verdana
color=red>"+res+"</center></font>");
            }%>


                                                                        <table align="center" width="40" >
                                    <tr><td><b>Patient&nbsp;ID</b></td><td><input type="text"
name="t1" style="font-family: Comic Sans MS" size="20"
value="<%=session.getAttribute("user").toString()%>" readonly/></td></tr>
```

```
                                   <tr><td><b>Random&nbsp;No</b></td><td><input type="text"
name="t2" style="font-family: Comic Sans MS" size="20"
value="<%=session.getAttribute("random").toString()%>" readonly/></td></tr>


                         <tr><td><b>Current&nbsp;BP</b></td><td><input type="text"
name="t3" style="font-family: Comic Sans MS" size="15"/></td></tr>


                         <tr><td><b>Current&nbsp;Heart&nbsp;Rate</b></td><td><input
type="text" name="t4" style="font-family: Comic Sans MS" size="15"/></td></tr>


                        <tr><td><b>Current&nbsp;Problem</b></td><td><input type="text"
name="t5" style="font-family: Comic Sans MS" size="35"/></td></tr>



                                  <tr><td></td><td><input type="submit" value="Submit"></td>

                                  </table>

                                            </div> </div>

</body>

</html>
```

3.2.10 PatientScreen.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div class="main">

  <div class="main_resize">

    <div class="header">
    <div class="logo">
     <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
    </div>
   </div>
   <div class="content">
    <div class="content_bg">
      <div class="menu_nav">
       <ul>
        <li class="active"><a href="PatientDataUpload.jsp">Patient Data Upload
Phase</a></li>
        <li><a href="CheckupPhase.jsp">Checkup Phase</a></li>
        <li><a href="Logout.jsp">Logout</a></li>
       </ul>
      </div>
      <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                <%
        String res = request.getParameter("t1");
        if(res != null){
                out.println("<center><font face=verdana
color=red>"+res+"</center></font>");
        }%>
     <p align="justify"><font size="3" style="font-family: Comic Sans MS">Abstract-
With the fast progress of network communication, its technologies and the developing
popularity of telecare medical information system (TMIS), doctors provide treatment to
patients via Internet without visiting hospitals. By using mobile device, wireless body
area network and cloud based architecture, the patients can gather their physiological
information and upload to cloud via their mobile devices. The authenticated doctor
provides online treatment to patient at anytime and anywhere.</p>  <p
align="justify"><font size="3" style="font-family: Comic Sans MS">Moreover, TMIS
maintains security and privacy of the patients in information communication and
authenticated to all the participants before assessing this system. Recently Li et al. (2018)
presented a cloud-assisted authentication and privacy preservation scheme for TMIS.
```

```
They believed that their scheme secure against all well-known privacy and security
attributes. In the proposed work, we reviewed Li et al. authentication protocol and found
that it has various security flaws like as message authentication fails in healthcare center
upload phase, session key is not possible in healthcare center upload phase,
impersonation attack in patient data upload phase, patient anonymity and patient
unlinkability.</p>

  </body>

</body>

</html>

</body>

</html>
```

3.2.11 Prescription.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

<script language="javascript">

        function validate(formObj)

        {

        if(formObj.t3.value.length==0)

        {

        alert("Please Enter Prescription");

        formObj.t3.focus();

        return false;

        }


        formObj.actionUpdateData.value="update";

        return true;

        }
```

```
          </script>
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
        <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
          <ul>
           <li class="active"><a href="Treatment.jsp">Treatment Phase</a></li>
                              <li><a href="Logout.jsp">Logout</a></li>
          </ul>
        </div>
        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                                          <center>
<form name="f1" method="post" action="Prescription" onsubmit="return
validate(this);"><br/>
   <h2><b>Patient Login Screen</b></h2>


          <%
          String res = request.getParameter("t1");
          if(res != null){
                out.println("<center><font face=verdana
color=red>"+res+"</center></font>");

        }%>


                                                    <table align="center" width="40" >

                                <tr><td><b>Record&nbsp;No</b></td><td><input type="text"
name="t1" style="font-family: Comic Sans MS" size="20"
value="<%=request.getParameter("t11").toString()%>" readonly/></td></tr>


                                <tr><td><b>Patient&nbsp;ID</b></td><td><input type="text"
name="t2" style="font-family: Comic Sans MS" size="20"
value="<%=request.getParameter("t2").toString()%>" readonly/></td></tr>


                          <tr><td><b>Prescription</b></td><td><input type="text" name="t3"
style="font-family: Comic Sans MS" size="45"/></td></tr>

                         <tr><td></td><td><input type="submit" value="Submit"></td>

                               </table>

                </div> </div>

        </body>

</html>
```

3.2.12 RegisterDoctor.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

<script language="javascript">

        function validate(formObj)

        {

        if(formObj.t1.value.length==0)
        {
        alert("Please Enter Username");
        formObj.t1.focus();
        return false;
        }
        if(formObj.t2.value.length==0)
        {
        alert("Please Enter Password");
        formObj.t2.focus();
        return false;
        }
        if(formObj.t4.value.length==0)
        {
        alert("Please Enter Contact No");
        formObj.t4.focus();
        return false;
        }
        if(formObj.t5.value.length==0)
        {
        alert("Please Enter Address");
        formObj.t5.focus();
        return false;
        }
        formObj.actionUpdateData.value="update";
        return true;
        }
        </script>
</head>
<body>
```

```
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
       <h1><span><center>PSMPA: Patient Self-Controllable
and Multi-Level Privacy-Preserving
Cooperative</center><center>Authentication in Distributed
m-Healthcare Cloud Computing System</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
          <ul>
            <li class="active"><a href="RegisterPatients.jsp">Healthcare Patient Upload
Phase</a></li>
            <li><a href="RegisterDoctor.jsp">Register Doctor</a></li>
                        <li><a href="ViewPatients.jsp">View Patient Details</a></li>
                        <li><a href="ViewDoctors.jsp">View Doctor Details</a></li>


            <li><a href="Logout.jsp">Logout</a></li>
          </ul>
        </div>
        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                                        <center>
<form name="f1" method="post" action="RegisterDoctor" onsubmit="return
validate(this);"><br/>
   <h2><b>Add Doctor Screen</b></h2>
```

```
        <%
        String res = request.getParameter("t1");
        if(res != null){
                out.println("<center><font face=verdana
color=red>"+res+"</center></font>");
        }%>
                        <table align="center" width="40" >
                         <tr><td><b>Username</b></td><td><input type="text"
name="t1" style="font-family: Comic Sans MS" size="20"/></td></tr>


                  <tr><td><b>Password</b></td><td><input type="password" name="t2"
style="font-family: Comic Sans MS" size="20"/></td></tr>


                   <tr><td><b>User Type</b></td><td><select name="t3">
                        <option value="Doctor">Doctor</option>
                        </select>
                        </td></tr>


                         <tr><td><b>Contact No</b></td><td><input type="text"
name="t4" style="font-family: Comic Sans MS" size="20"/></td></tr>
  <tr><td><b>Address</b></td><td><input type="text" name="t5" style="font-family:
Comic Sans MS" size="50"/></td></tr><tr><td><b>Qualification</b></td><td><select
name="t6">
                        <option value="MBBS">MBBS</option>
                        <option value="BUMS">BUMS</option>
                        </select>
                        </td></tr>


                         <tr><td><b>Speciality</b></td><td><select name="t7">
                        <option value="HeartSpecialist">HeartSpecialist</option>
                        <option value="CancerSpecialist">CancerSpecialist</option>
                        <option value="Others">Others</option>
                        </select>
                        </td></tr>



                        <tr><td></td><td><input type="submit" value="Add
Doctor"></td>
                        </table>
                                </div>
          </div></body>
</html>
```

3.2.13 RegisterPatients.jsp

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
<script language="javascript">
        function validate(formObj)
        {
        if(formObj.t1.value.length==0)
        {
        alert("Please Enter patient name");
        formObj.t1.focus();
        return false;
        }
        if(formObj.t2.value.length==0)
```

```
{
alert("Please Enter birth date");
formObj.t2.focus();
return false;
}
if(formObj.t5.value.length==0)
{
alert("Please Enter Contact No");
formObj.t5.focus();
return false;
}
if(formObj.t6.value.length==0)
{
alert("Please Enter Address");
formObj.t6.focus();
return false;
}
if(formObj.t7.value.length==0)
{
alert("Please Enter problem");
formObj.t7.focus();
return false;
}
if(formObj.t8.value.length==0)
{
alert("Please Enter problem description");
formObj.t8.focus();
return false;
}
```

```
          formObj.actionUpdateData.value="update";
          return true;
          }
          </script>
           <script language="javascript" type="text/javascript" src="datetimepicker.js">
</script>
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
       <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
          <ul>
            <li class="active"><a href="RegisterPatients.jsp">Healthcare Patient Upload
Phase</a></li>
            <li><a href="RegisterDoctor.jsp">Register Doctor</a></li>
                        <li><a href="ViewPatients.jsp">View Patient Details</a></li>
                        <li><a href="ViewDoctors.jsp">View Doctor Details</a></li>


            <li><a href="Logout.jsp">Logout</a></li>
          </ul>
        </div>
    <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>

                                        <center>

<form name="f1" method="post" action="HealthcareUpload" onsubmit="return
validate(this);"><br/>

  <h2><b>Patient Profile Screen</b></h2>



          <%

          String res = request.getParameter("t1");

          if(res != null){

                    out.println("<center><font face=verdana
color=red>"+res+"</center></font>");

          }%>



                                                  <table align="center" width="40" >

                              <tr><td><b>Patient&nbsp;Name</b></td><td><input
type="text" name="t1" style="font-family: Comic Sans MS" size="20"/></td></tr>



                       <tr><td><b>Birth Date</b></td><td><input name="t2" type="Text"
id="demo1" maxlength="25" size="20" class="c2" ><a
href="javascript:NewCal('demo1','ddmmmyyyy',true,24)"><img src="cal.gif" width="16"
height="16" border="0" alt="Pick a date"></a>

                              <span class="descriptions"></span></td></tr>



                        <tr><td><b>Age</b></td><td><select name="t3">

                              <%for(int i=1;i<=200;i++){%>

                              <option value="<%=i%>"><%=i%></option>

                              <%}%>

                              </select>

                              </td></tr>



                               <tr><td><b>Gender</b></td><td><select name="t4">
                        <option value="Male">Male</option>

                        <option value="Female">Female</option>

                        </select>

                        </td></tr>

<tr><td><b>Contact&nbsp;No</b></td><td><input type="text" name="t5" style="font-family: Comic Sans MS" size="20"/></td></tr>


                         <tr><td><b>Address</b></td><td><input type="text" name="t6" style="font-family: Comic Sans MS" size="50"/></td></tr>
<tr><td><b>Problem</b></td><td><input type="text" name="t7" style="font-family: Comic Sans MS" size="80"/></td></tr>
<tr><td><b>Problem&nbsp:Description</b></td><td><textarea name="t8" cols="60" rows="8"></textarea></td></tr>

              <tr><td></td><td><input type="submit" value="Create Profile"></td>

                        </table>

                                </div>

                                </div>

        </body>

</html>
```

3.2.14 Treatment.jsp

```
<%@page import="java.sql.Connection"%>

<%@page import="java.sql.ResultSet"%>

<%@page import="java.sql.Statement"%>

<%@page import="com.DBConnection"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

<script language="javascript">
```

```
          function validate(formObj)
          {
          if(formObj.t1.value.length==0)
          {
          alert("Please Enter Username");
          formObj.t1.focus();
          return false;
          }
          if(formObj.t2.value.length==0)
          {
          alert("Please Enter Password");
          formObj.t2.focus();
          return false;
          }
          if(formObj.t3.value.length==0)
          {
          alert("Please Enter Random Number");
          formObj.t3.focus();
          return false;
          }
          formObj.actionUpdateData.value="update";
          return true;
          }
          </script>
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
```

```
      <div class="logo">
       <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
          <ul>
           <li class="active"><a href="Treatment.jsp">Treatment Phase</a></li>
                              <li><a href="Logout.jsp">Logout</a></li>
          </ul>
        </div>
        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                                          <center>
<form name="f1" method="post" action="TreatmentPage.jsp" onsubmit="return
validate(this);"><br/>
   <h2><b>Treatment Phase Screen</b></h2>


           <%
           String res = request.getParameter("t1");
           if(res != null){
                      out.println("<center><font face=verdana
color=red>"+res+"</center></font>");
           }%>


                                                        <table align="center" width="40" >
                              <tr><td><b>Patient&nbsp;ID</b></td><td>
                             <select name="t1">
```

```
                              <%
          Connection con = DBConnection.getCon();
          String pid = session.getAttribute("user").toString();
          Statement stmt = con.createStatement();
          ResultSet rs = stmt.executeQuery("select patient_id from patientprofile");
          while(rs.next()){
                    String str = rs.getString(1);%>
                    <option value="<%=str%>"><%=str%></option>
                    <%}%>
                              </select>
                              </td></tr>
       <tr><td></td><td><input type="submit" value="Submit"></td>
                              </table>
                                        </div>
                              </div>
                    </body>
</html>
```

3.2.15 TreatmentPhase.jsp

```
<%@page import="java.util.ArrayList"%>
<%@page import="java.io.ObjectOutputStream"%>
<%@page import="java.io.ObjectInputStream"%>
<%@page import="java.io.File"%>
<%@page import="java.io.FileInputStream"%>
<%@page import="com.ECC"%>
<%@page import="com.Point"%>
<%@page import="com.DBConnection"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
```

```
<title></title>

<meta http-equiv="content-type" content="text/html; charset=utf-8" />

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div class="main">

  <div class="main_resize">

    <div class="header">

      <div class="logo">

       <h1><span><center>A secure elliptic curve cryptography based mutual
authentication

protocol </center><center>for cloud-assisted</center> </span><small></small></h1>

      </div>

    </div>

    <div class="content">

      <div class="content_bg">

        <div class="menu_nav">

        <ul>

            <li class="active"><a href="Treatment.jsp">Treatment Phase</a></li>

                                <li><a href="Logout.jsp">Logout</a></li>

          </ul>

        </div>

        <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>

                                        <center>


   <h2><b>View Patient Disease Details Screen</b></h2>


            <%

            String res = request.getParameter("t1");
        if(res != null){

                out.println("<center><font face=verdana
color=red>"+res+"</center></font>");

        }%>

                        <table border="1" align="center" width="100%">

                        <tr><th><font size="3" color="black">Record No</th>

                        <tr><th><font size="3" color="black">Patient ID</th><th><font
size="3" color="black">Random No</th><th><font size="3" color="black">BP</th>

                        <th><font size="3" color="black">Heart Rate</th><th><font
size="3" color="black">Problem</th>

                        <th><font size="3" color="black">Give Prescription</th>

                        <tr>

        <%

        String pid = request.getParameter("t1").toString();

        String random = DBConnection.getRandom(pid);

        String path = getServletContext().getRealPath("/")+"WEB-
INF/patients/"+pid+".txt";

        File file = new File(path);

        if(file.exists()) {

                ECC ecc = new ECC();

                ecc.loadKeys(random,random,random);

                ObjectInputStream oin = new ObjectInputStream(new
FileInputStream(file));

                Object obj = (Object)oin.readObject();

                ArrayList<Point[]> list = (ArrayList<Point[]>)obj;

                oin.close();

                for(int i=0;i<list.size();i++){

                        Point decrypt = ecc.getKeys().decrypt(list.get(i));

                        byte b1[] = decrypt.getX().toByteArray();

                        byte b2[] = decrypt.getY().toByteArray();

                        String arr[] = new String(b1).split(",");
                        String str = DBConnection.getPrescription(pid,i+"");
                        if(str.equals("none")){
                %>
        <tr><td><font size="3" color="black"><%=pid%></td>
        <td><font size="3" color="black"><%=random%></td>
        <td><font size="3" color="black"><%=arr[0]%></td>
        <td><font size="3" color="black"><%=arr[1]%></td>
        <td><font size="3" color="black"><%=arr[2]%></td>
        <td><a href="Prescription.jsp?t11=<%=i%>&t2=<%=pid%>">
        <font size="3" color="black">Click Here</a></td>
        <%}}}%>
        </tr>
        </table>
        </body>
</html>
```

3.2.16 ViewDoctors.jsp

```
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="com.DBConnection"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div class="main">
```

```
 <div class="main_resize">
   <div class="header">
     <div class="logo">
      <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
     </div>
   </div>
   <div class="content">
     <div class="content_bg">
       <div class="menu_nav">
        <ul>
          <li class="active"><a href="RegisterPatients.jsp">Healthcare Patient Upload
Phase</a></li>
          <li><a href="RegisterDoctor.jsp">Register Doctor</a></li>
                              <li><a href="ViewPatients.jsp">View Patient Details</a></li>
                              <li><a href="ViewDoctors.jsp">View Doctor Details</a></li>


          <li><a href="Logout.jsp">Logout</a></li>
         </ul>
       </div>
       <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>
                                        <center>
 <h2><b>View Physician Details Screen</b></h2>
  <%
          String res = request.getParameter("t1");
          if(res != null){
                    out.println("<center><font face=verdana
color=red>"+res+"</center></font>");
          }%>
```

```
                        <table border="1" align="center" width="50%">

                        <tr><th><font size="3" color="black">UserName</th><th><font size="3" color="black">Password</th><th><font size="3" color="black">User Type</th>

                        <th><font size="3" color="black">Contact No</th><th><font size="3" color="black">Address</th><th><font size="3" color="black">Qualification</th>

                        <th><font size="3" color="black">Speciality</th><th><font size="3" color="black">Random No</th>

                        <th><font size="3" color="black">Public/Secret Key</th>

                        <tr>

        <%

        Connection con = DBConnection.getCon();

        String pid = session.getAttribute("user").toString();

        Statement stmt = con.createStatement();

        ResultSet rs = stmt.executeQuery("select * from adddoctor");

        while(rs.next()){

                String str = rs.getString(9);

                str = str.substring(0,30);

                %>

        <tr><td><font size="3" color="black"><%=rs.getString(1)%></td>

        <td><font size="3" color="black"><%=rs.getString(2)%></td>

        <td><font size="3" color="black"><%=rs.getString(3)%></td>

        <td><font size="3" color="black"><%=rs.getString(4)%></td>

        <td><font size="3" color="black"><%=rs.getString(5)%></td>

        <td><font size="3" color="black"><%=rs.getString(6)%></td>

        <td><font size="3" color="black"><%=rs.getString(7)%></td>

        <td><font size="3" color="black"><%=rs.getString(8)%></td>

        <td><font size="3" color="black"><%=str%></td>

        <%}%>

        </tr>
        </table>
        </body>
</html>
```

3.2.17 ViewPatients.jsp

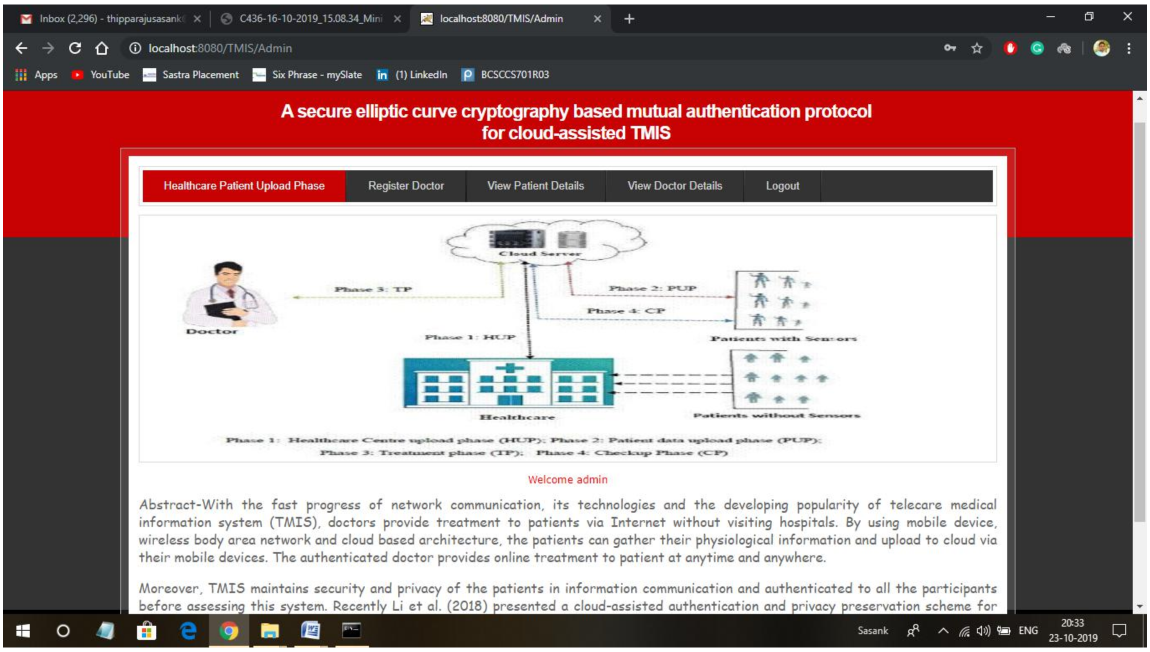
```
<%@page import="java.sql.Connection"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.Statement"%>
<%@page import="com.DBConnection"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title></title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div class="main">
  <div class="main_resize">
    <div class="header">
      <div class="logo">
       <h1><span><center>A secure elliptic curve cryptography based mutual
authentication
protocol </center><center>for cloud-assisted</center> </span><small></small></h1>
      </div>
    </div>
    <div class="content">
      <div class="content_bg">
        <div class="menu_nav">
         <ul>
```

```
          <li class="active"><a href="RegisterPatients.jsp">Healthcare Patient Upload
Phase</a></li>

          <li><a href="RegisterDoctor.jsp">Register Doctor</a></li>

                         <li><a href="ViewPatients.jsp">View Patient Details</a></li>

                         <li><a href="ViewDoctors.jsp">View Doctor Details</a></li>


          <li><a href="Logout.jsp">Logout</a></li>

        </ul>

      </div>

      <div class="hbg"><img src="images/header_images.JPG" width="915"
height="286" alt="" /></div>

                                   <center>


  <h2><b>View Patients Details Screen</b></h2>


          <%

          String res = request.getParameter("t1");

          if(res != null){

                    out.println("<center><font face=verdana
color=red>"+res+"</center></font>");

          }%>

                         <table border="1" align="center" width="100%">

                         <tr><th><font size="3" color="black">Patient ID</th><th><font
size="3" color="black">Patient Name</th><th><font size="3" color="black">Birth
Date</th>

                         <th><font size="3" color="black">Age</th><th><font size="3"
color="black">Gender</th><th><font size="3" color="black">Contact No</th>

                         <th><font size="3" color="black">Address</th><th><font
size="3" color="black">Problem</th>

                         <th><font size="3" color="black">Problem Description</th>

                         <th><font size="3" color="black">Random No</th>

                         <th><font size="3" color="black">Public/Secret Key</th>
```

```
                        <tr>
          <%
          Connection con = DBConnection.getCon();
          String pid = session.getAttribute("user").toString();
          Statement stmt = con.createStatement();
          ResultSet rs = stmt.executeQuery("select * from patientprofile");
          while(rs.next()){
                    String str = rs.getString(11);
                    str = str.substring(0,30);%>
          <tr><td><font size="3" color="black"><%=rs.getString(1)%></td>
          <td><font size="3"
color="black"><%=DBConnection.anonymity(rs.getString(2))%></td>
          <td><font size="3" color="black"><%=rs.getString(3)%></td>
          <td><font size="3" color="black"><%=rs.getString(4)%></td>
          <td><font size="3" color="black"><%=rs.getString(5)%></td>
          <td><font size="3" color="black"><%=rs.getString(6)%></td>
          <td><font size="3" color="black"><%=rs.getString(7)%></td>
          <td><font size="3" color="black"><%=rs.getString(8)%></td>
          <td><font size="3" color="black"><%=rs.getString(9)%></td>
          <td><font size="3" color="black"><%=rs.getString(10)%></td>
          <td><font size="3" color="black"><%=str%></td>
          <%}%>
          </tr>
          </table>
          </body>
</html>
```

CHAPTER 4

SNAPSHOTS


Inbox (2,296) - thipparajusasank
C436-16-10-2019_15.08.34_Mini
localhost:8080/TMIS/Admin
localhost:8080/TMIS/Admin
Apps
YouTube
Sastra Placement
Six Phrase - mySlate
(1) LinkedIn
BCSCCS701R03
Healthcare Patient Upload Phase
Register Doctor
View Patient Details
View Doctor Details
Logout
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)
Sasank
ENG
20:33
23-10-2019

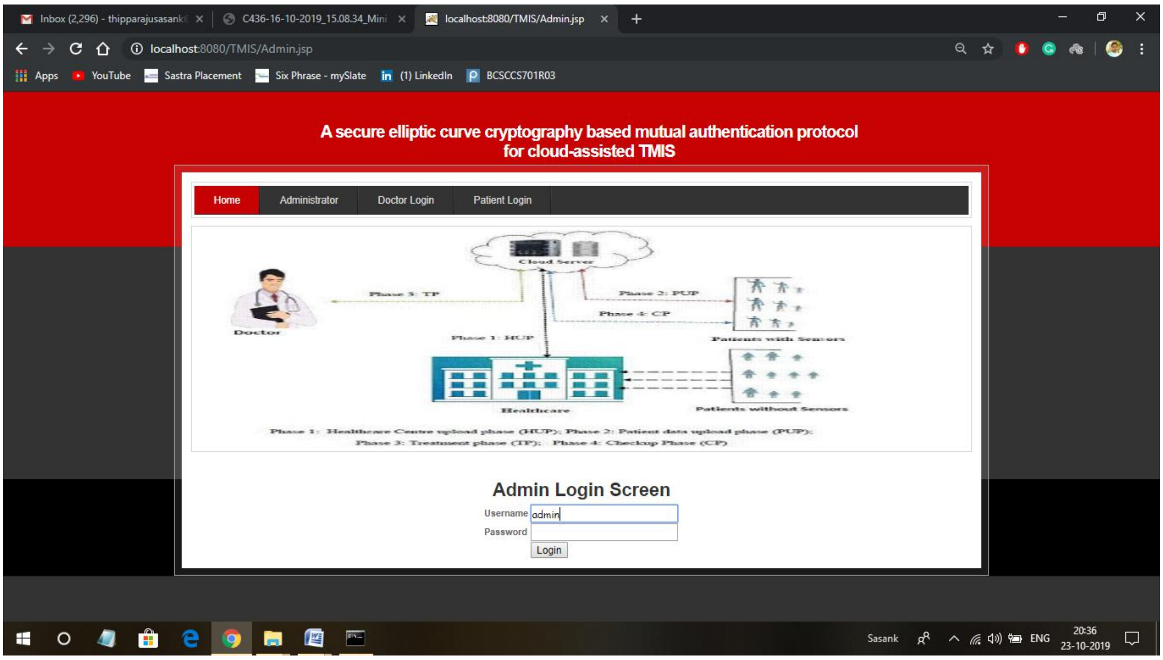
A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS

Welcome admin

Abstract-With the fast progress of network communication, its technologies and the developing popularity of telecare medical information system (TMIS), doctors provide treatment to patients via Internet without visiting hospitals. By using mobile device, wireless body area network and cloud based architecture, the patients can gather their physiological information and upload to cloud via their mobile devices. The authenticated doctor provides online treatment to patient at anytime and anywhere.

Moreover, TMIS maintains security and privacy of the patients in information communication and authenticated to all the participants before assessing this system. Recently Li et al. (2018) presented a cloud-assisted authentication and privacy preservation scheme for

THIS IS HOME PAGE


Inbox (2,296) - thipparajusasank
C436-16-10-2019_15.08.34_Mini
localhost:8080/TMIS/Admin.jsp
localhost:8080/TMIS/Admin.jsp
Apps
YouTube
Sastra Placement
Six Phrase - mySlate
(1) LinkedIn
BCSCCS701R03
Home
Administrator
Doctor Login
Patient Login
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)
Login
Sasank
ENG
20:36
23-10-2019

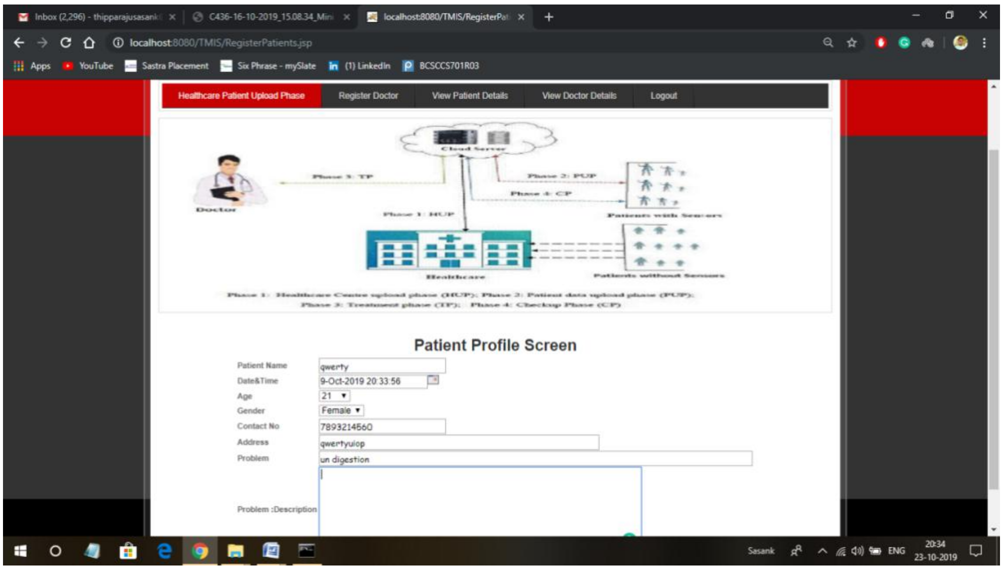
A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS

Admin Login Screen

Username admin

Password

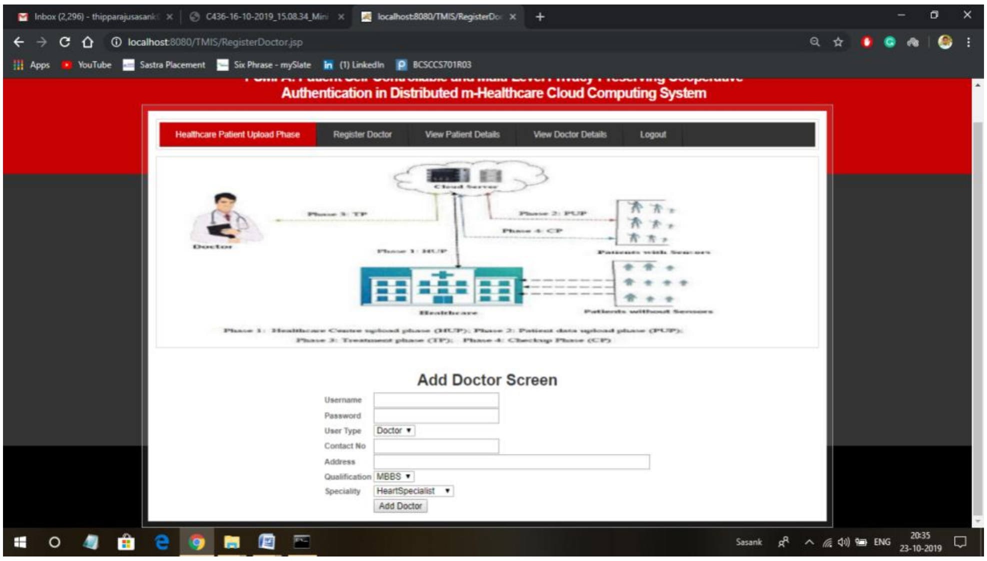
THIS ADMIN’S LOGIN PAGE


localhost:8080/TMIS/RegisterPatients.jsp
Healthcare Patient Upload Phase
Register Doctor
View Patient Details
View Doctor Details
Logout
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)

Patient Profile Screen

Patient Name	qwerty
Date&Time	9-Oct-2019 20:33:56
Age	21
Gender	Female
Contact No	7893214560
Address	qwertyuiop
Problem	un digestion
Problem :Description	

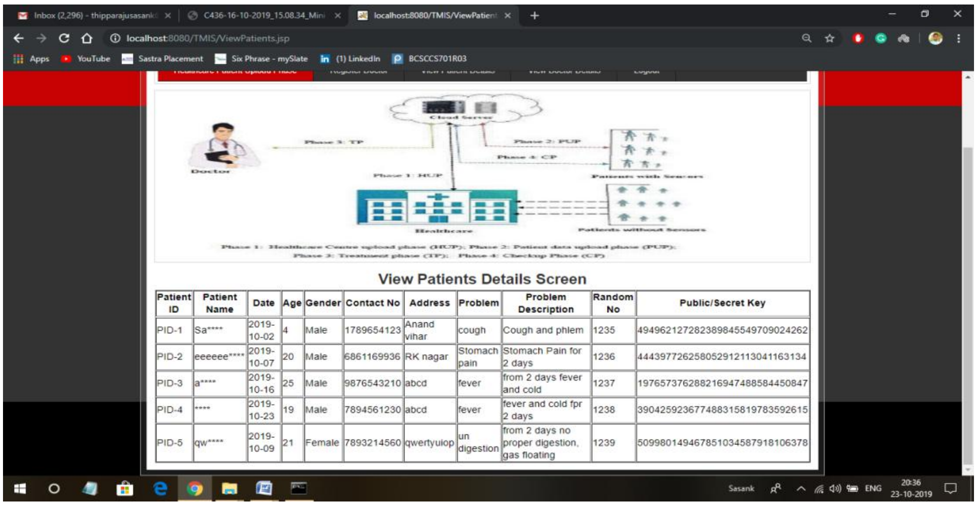
HERE HEALTHCENTER REGISTERS THE PATIENT DETAILS


localhost:8080/TMIS/RegisterDoctor.jsp
Authentication in Distributed m-Healthcare Cloud Computing System
Healthcare Patient Upload Phase
Register Doctor
View Patient Details
View Doctor Details
Logout
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)
Add Doctor

Add Doctor Screen

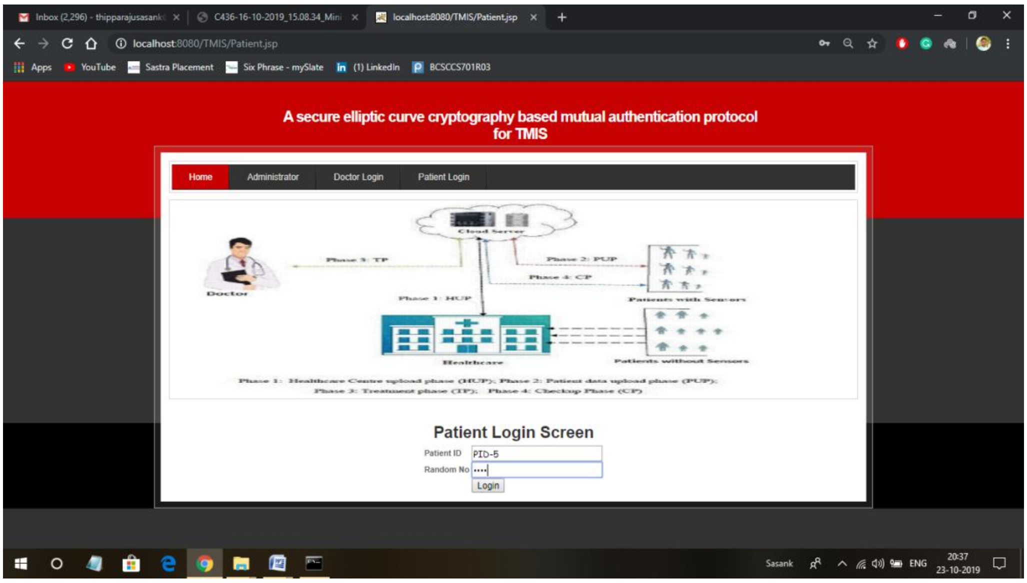
Username	
Password	
User Type	Doctor
Contact No	
Address	
Qualification	MBBS
Speciality	HeartSpecialist

HERE HEALTHCENTER ADD'S THE DOCTOR


localhost:8080/TMIS/ViewPatients.jsp
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)

View Patients Details Screen

Patient ID	Patient Name	Date	Age	Gender	Contact No	Address	Problem	Problem Description	Random No	Public/Secret Key
PID-1	Sa****	2019-10-02	4	Male	1789654123	Anand vihar	cough	Cough and phlem	1235	4949621272823898455497090242 62
PID-2	eeeeee****	2019-10-07	20	Male	6861169936	RK nagar	Stomach pain	Stomach Pain for 2 days	1236	44439772625805291211304116313 4
PID-3	a****	2019-10-16	25	Male	9876543210	abcd	fever	from 2 days fever and cold	1237	197657376288216947488584450847
PID-4	****	2019-10-23	19	Male	7894561230	abcd	fever	fever and cold fpr 2 days	1238	390425923677488315819783592615
PID-5	qw****	2019-10-09	21	Female	7893214560	qwertyuiop	un digestion	from 2 days no proper digestion, gas floating	1239	509980149467851034587918106378


localhost:8080/TMIS/Patient.jsp
Home
Administrator
Doctor Login
Patient Login
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)
Login

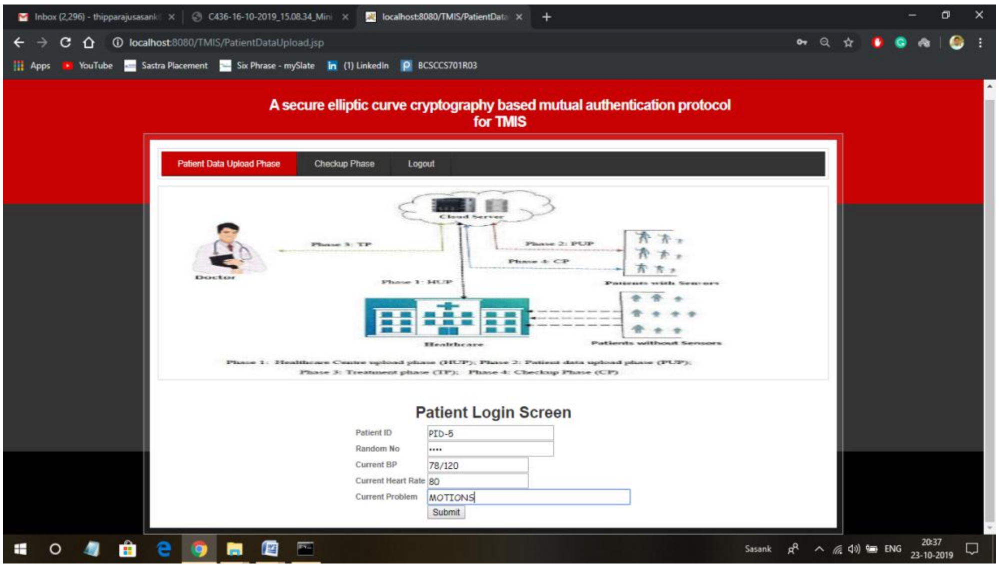
A secure elliptic curve cryptography based mutual authentication protocol for TMIS

Patient Login Screen

Patient ID PID-5

Random No ····

PATIENT LOGIN PAGE


localhost:8080/TMIS/PatientDataUpload.jsp
Patient Data Upload Phase
Checkup Phase
Logout
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)
Submit

A secure elliptic curve cryptography based mutual authentication protocol for TMIS

Patient Login Screen

Patient ID PID-5

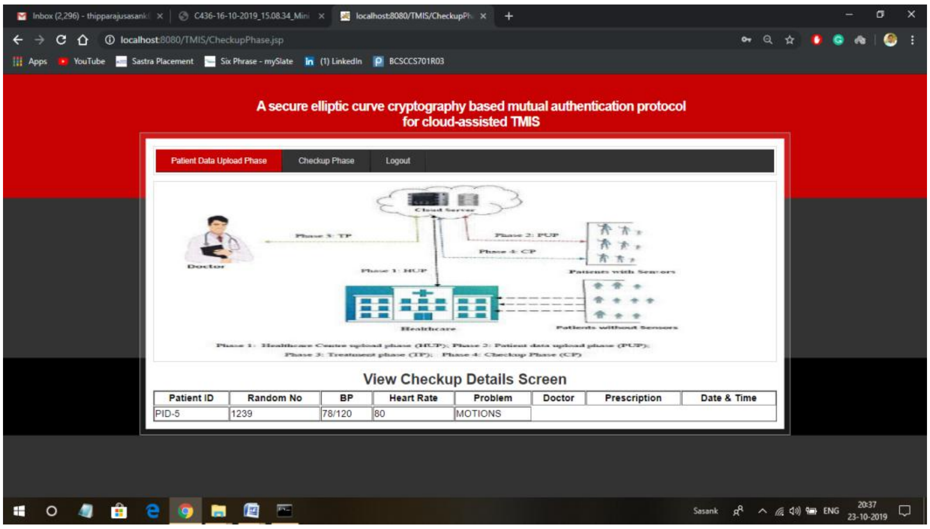
Random No ····

Current BP 78/120

Current Heart Rate 80

Current Problem MOTIONS

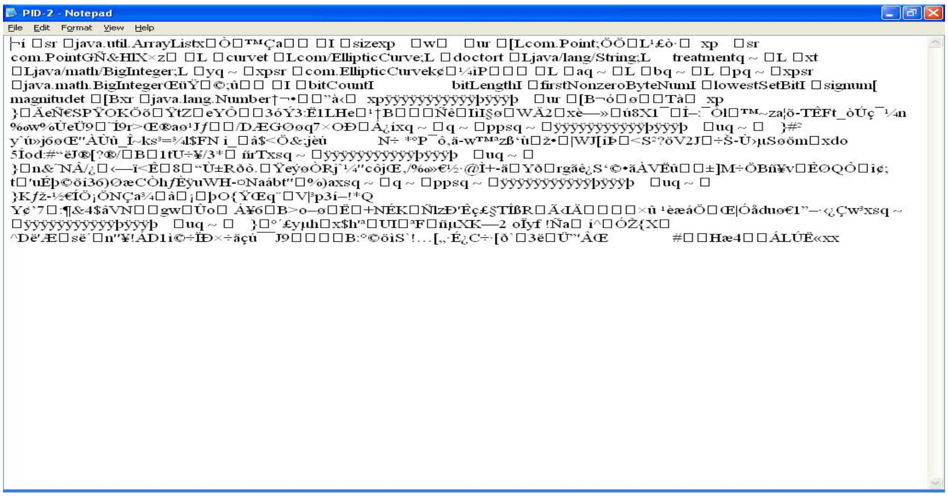
HERE WILL PATIENT UPLOAD HIS PROBLEM


localhost:8080/TMIS/CheckupPhase.jsp
Patient Data Upload Phase
Checkup Phase
Logout
Cloud Server
Doctor
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)

A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS

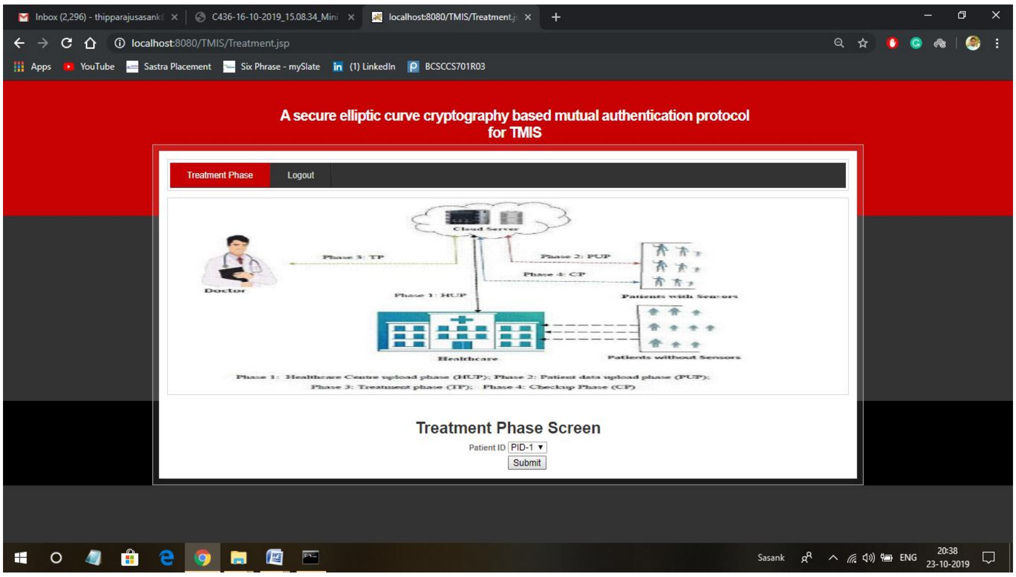
View Checkup Details Screen

Patient ID	Random No	BP	Heart Rate	Problem	Doctor	Prescription	Date & Time
PID-5	1239	78/120	80	MOTIONS			



```
├í □sr □java.util.ArrayListx□Ò□™Ça□□ □I □sizexp  □w□  □ur □[Lcom.Point;ÖÖ□L¹£ò·□  xp  □sr
com.PointGÑ&HIX×z□ □L □curvet □Lcom/EllipticCurve;L □doctort □Ljava/lang/String;L   treatmentq ~ □L □xt
□Ljava/math/BigInteger;L □yq ~ □xpsr □com.EllipticCurvek¢□¼iP□□□ □L □aq ~ □L □bq ~ □L □pq ~ □xpsr
□java.math.BigIntegerŒüŸ□©;û□□ □I □bitCountI          bitLengthI □firstNonzeroByteNumI □lowestSetBitI □signum[
magnitudet □[Bxr □java.lang.Number†¬•□□"à‹□  xpÿÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ  □ur □[B¬ó□ø□□Tà□  xp
}□ÄeÑ€SPŸOKÖõ□ŸtZ□eYÔ□□3óÝ3:Ë1LHe□¹†B□□□Ñë□IiI§ø□WÄ2□xè—»□ú8X1¯□Í–:¯Ò1□™~za¦õ-TÊFt_òÚç¯¼n
‰w‰ÙeÜ9□˜Ï9r>Œ®aø¹Jƒ□□/DÆGØøq7×OÐ□Á¿íxq ~ □q ~ □ppsq ~ □ÿÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ  □uq ~ □  }#²
y˜ú»j6øŒ"ÀÙû_Î~ks³=¾l$FN i_□â$<Ö&;jèú        N÷ *°P¯ô,ä-w™ªzß'ù□ž•□|WJ[íÞ□<S²?öV2J□÷Š-Ù›µSøöm□xdo
5Íod:#"ëJ®[?®/□B□1tU÷¥/3*□ ñrTxsq ~ □ÿÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ  □uq ~ □
}□n&˜NÂ/¿□‹—ï<Ê□8□"Ù±Rðô.□ŸeýøÒRj˜¼"côjŒ,/‰»€½·@Ì+-ä□Yð□rgãê¿S'©•ãÀVËû□□±]M÷ÖBñ¥v□ÊØQÔ□ì¢;
t□'uÉþ©öi36)ØæCÒhƒÈÿuWH-¤Naábt"□%)axsq ~ □q ~ □ppsq ~ □ÿÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ  □uq ~ □
}Kƒž-½€ÍÖ¡ÖNÇa¾□â□¡□þO{ŸŒq¨□V|³p3í–!*Q
Y¢'7□:¶&4$âVN□□gw□Ûo□ Á¥6□B>o–ø□Ë□+NÉK□ÑlzÐ'Êç£§TÍßR□Ã‹IÄ□□□□×û ¹èæáÖ□Œ|Óåduø€1"–·‹¿Çw³xsq ~
□ÿÿÿÿÿÿÿÿÿÿÿÿþÿÿÿþ  □uq ~ □  }□°´£yµh□x$h'ª□UI□ªF□ñµXK—2 oÏyf !Ña□ í^□ÓŽ{X□
^Dë'Æ□së´□n"¥!ÁD1i©÷ÏÐ×÷äçú¯J9□□□□B:°©öiS`!…[„·É¿C÷·[ð`□3ë□Ü"ÂŒ        #□□Hæ4□□ÁLÚË«xx
```

PATIENT DATA IS ENCRYPTED

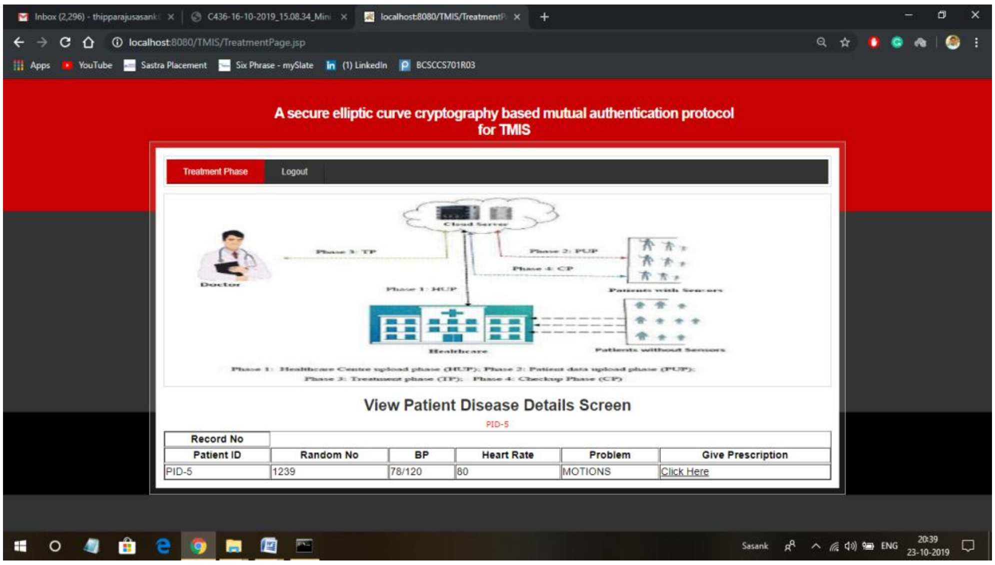

Inbox (2,296) - thipparajusasank
C436-16-10-2019_15.08.34_Mini
localhost:8080/TMIS/Treatment.js
localhost:8080/TMIS/Treatment.jsp
Apps
YouTube
Sastra Placement
Six Phrase - mySlate
(1) LinkedIn
BCSCCS701R03
Treatment Phase
Logout
Cloud Server
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Doctor
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)
Submit
Sasank
ENG
20:38
23-10-2019

A secure elliptic curve cryptography based mutual authentication protocol for TMIS

Treatment Phase Screen

Patient ID PID-1

DOCTOR GETS LOGIN AND CHECKS PATIENT'S PROBLEM


Inbox (2,296) - thipparajusasank
C436-16-10-2019_15.08.34_Mini
localhost:8080/TMIS/TreatmentP
localhost:8080/TMIS/TreatmentPage.jsp
Apps
YouTube
Sastra Placement
Six Phrase - mySlate
(1) LinkedIn
BCSCCS701R03
Treatment Phase
Logout
Cloud Server
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Phase 1: HUP
Doctor
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)
Sasank
ENG
20:39
23-10-2019

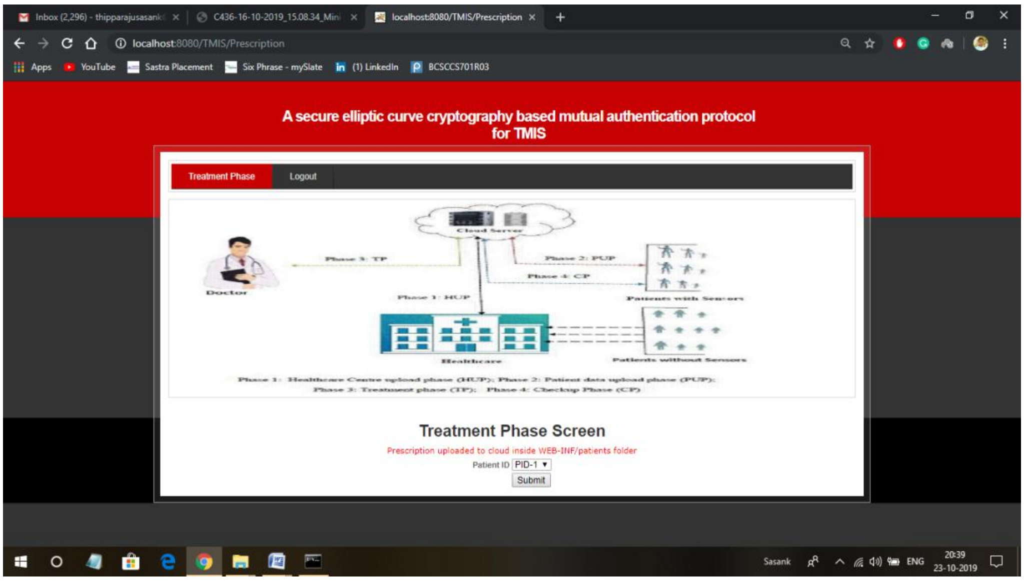
A secure elliptic curve cryptography based mutual authentication protocol for TMIS

View Patient Disease Details Screen

PID-5

Record No					
Patient ID	Random No	BP	Heart Rate	Problem	Give Prescription
PID-5	1239	78/120	80	MOTIONS	Click Here

DOCTOR GIVES PRESCRIPTION FOR PATIENT'S HEALTH CONDITION


Inbox (2,296) - thipparajusasank
C436-16-10-2019_15.08.34_Mini
localhost:8080/TMIS/Prescription
localhost:8080/TMIS/Prescription
Apps
YouTube
Sastra Placement
Six Phrase - mySlate
(1) LinkedIn
BCSCCS701R03

A secure elliptic curve cryptography based mutual authentication protocol for TMIS

Treatment Phase
Logout
Cloud Server
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Doctor
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)

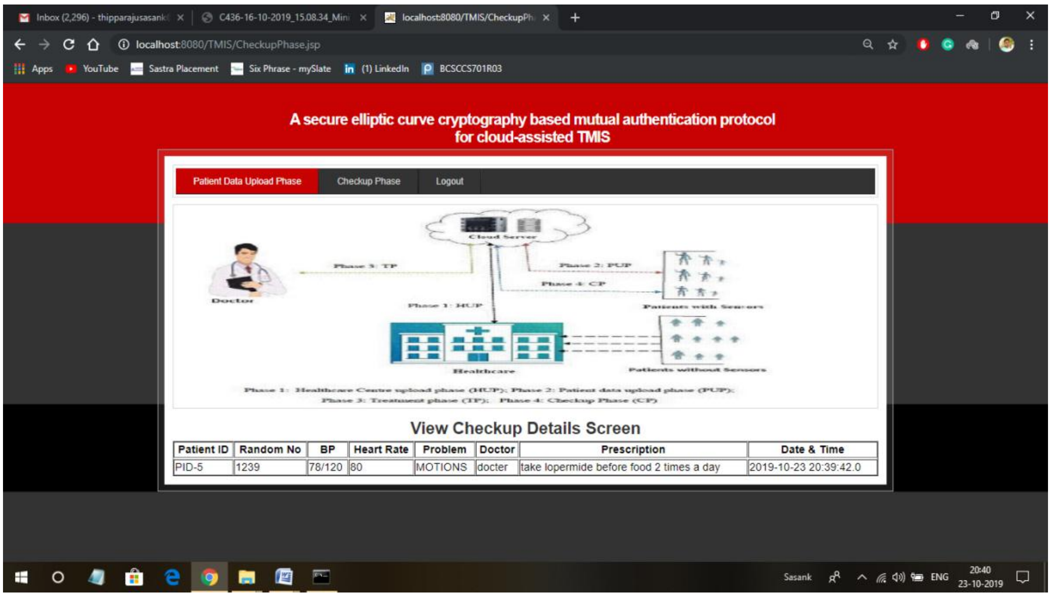
Treatment Phase Screen

Prescription uploaded to cloud inside WEB-INF/patients folder

Patient ID PID-1 ▾

Submit

Sasank
ENG
20:39
23-10-2019


Inbox (2,296) - thipparajusasank
C436-16-10-2019_15.08.34_Mini
localhost:8080/TMIS/CheckupPh
localhost:8080/TMIS/CheckupPhase.jsp
Apps
YouTube
Sastra Placement
Six Phrase - mySlate
(1) LinkedIn
BCSCCS701R03

A secure elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS

Patient Data Upload Phase
Checkup Phase
Logout
Cloud Server
Phase 3: TP
Phase 2: PUP
Phase 4: CP
Doctor
Phase 1: HUP
Patients with Sensors
Healthcare
Patients without Sensors
Phase 1: Healthcare Centre upload phase (HUP); Phase 2: Patient data upload phase (PUP);
Phase 3: Treatment phase (TP); Phase 4: Checkup Phase (CP)

View Checkup Details Screen

Patient ID	Random No	BP	Heart Rate	Problem	Doctor	Prescription	Date & Time
PID-5	1239	78/120	80	MOTIONS	docter	take lopermide before food 2 times a day	2019-10-23 20:39:42.0

Sasank
ENG
20:40
23-10-2019

PATIENT CAN VIEW HIS CHECKUP BY LOGGING IN

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

The paper is the approach of an enhanced secure and efficient elliptic curve cryptography based mutual authentication protocol for cloud-assisted TMIS. In this paper, we reviewed Li et al. scheme and found that, message authentication fails in health care center upload phase, session key is not possible in healthcare center upload phase, impersonation attack in patient data upload phase, patient anonymity patient un link ability and doctor unlink ability.Further, we proposed a new enhance protocol in same environment. The paper shows the security analysis of the proposed protocol which based on several security attributes and functionality features. Hence, the presented protocol manages the better security feature and attributes compare to their previous related protocols in TMIS. Furthermore, we show the performance of the proposed protocol which is efficient in the term of computation and communication cost in cloud based TMIS environment.

CHAPTER 6

REFERENCES

1. **C.-L. Chen, T.-F. Shih, T.-T. Yang.**

   a) A secure medical data exchange protocol based on cloud environment – J. Med. Syst., 38 (9) (2014), p. 112

   b) A privacy authentication scheme based on cloud for medical environment – J. Med. Syst., 38 (11) (2014), p. 143

2. **J. Liu, Y. Chiou, Z. Ying**

   Improvement of a privacy authentication scheme based on cloud for medical environment – J. Med. Syst., 40 (4) (2016), p. 10

3. **C.-C. Wang C.-T. Li, D.-H. Shih,**

   On the security of a privacy authentication scheme based on cloud for medical environment – International Conference on Information Science and Applications, Springer(2017), pp. 241-248

4. **A. Karati, G. Biswas, M.K. Khan, P. Mohit, R. Amin**

   A standard mutual authentication protocol for cloud computing based health care system – J. Med. Syst., 41 (4) (2017), p. 50

5. **M. Ahmad,** V. **Kumar, S. Jangirala**

   a.) An efficient mutual authentication framework for healthcare system in cloud computing – J. Med. Syst., 42 (8) (2018), p. 142.

   b) Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems –Compute. Methods Programs Biomed, 157 (2018), pp. 191-203.

6. **URL**: https://scialert.net/abstract/?doi=jas.2005.604.633

   For theory and implementation of ECC

7. Performance of RSA and ELLIPTIC CURVE CRYPTOGRAPHY

   URL:https://ieeexplore.ieee.org/document/7917979