Appendix A: Implementation codes

DES Algorithm

```
package com.auth;

import java.security.Key;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;

public class AESAlgorithm
{
	private static final String ALGORITHM = "AES";
	private static final int ITERATIONS = 2;
	private static final byte[] keyValue = new byte[] { 'T', 'h', 'i', 's', 'I', 's', 'A', 'S', 'e', 'c', 'r', 'e', 't', 'K',
'e', 'y' };

	public static String encrypt(String value, String salt) throws Exception
	{
		Key key = generateKey();
		Cipher c = Cipher.getInstance(ALGORITHM);
		c.init(Cipher.ENCRYPT_MODE, key);
		String valueToEnc = null;
		String eValue = value;
		for (int i = 0; i < ITERATIONS; i++)
		{
			valueToEnc = salt + eValue;
			byte[] encValue = c.doFinal(valueToEnc.getBytes());
			eValue = new BASE64Encoder().encode(encValue);
		}

			return eValue;
```

```
    }

    public static String decrypt(String value, String salt) throws Exception {
        Key key = generateKey();
        Cipher c = Cipher.getInstance(ALGORITHM);
        c.init(Cipher.DECRYPT_MODE, key);

        String dValue = "";
        String valueToDecrypt = value;
        for (int i = 0; i < ITERATIONS; i++) {
            byte[] decordedValue = new BASE64Decoder().decodeBuffer(valueToDecrypt);
            byte[] decValue = c.doFinal(decordedValue);
            dValue = new String(decValue).substring(salt.length());
            valueToDecrypt = dValue;
        }

        return dValue;
    }


    private static Key generateKey() throws Exception
    {
            Key key = new SecretKeySpec(keyValue, ALGORITHM);
            return key;
    }
    public static void main(String[] args)throws Exception {

            String str = "jaydeep";
            String enc = encrypt(str, "AES");
            String des = decrypt(enc, "AES");
            System.out.println(enc);
            System.out.println(des);
    }

}
```

Key Generation

```
package com.auth;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Random;

public class KeyGenerator {

   private static final String CHAR_LIST =
      "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890";
   private static final String NUM_LIST =
         "1234567890";
  // private static final int RANDOM_STRING_LENGTH = 5;

   /**
    * This method generates random string
    * @return
    */

   public String generateRandomString(int RANDOM_STRING_LENGTH){
       StringBuffer randStr = new StringBuffer();
      for(int i=0; i<RANDOM_STRING_LENGTH; i++){
         int number = getRandomNumber();
         char ch = CHAR_LIST.charAt(number);
         randStr.append(ch);
      }
      return randStr.toString();
   }
   public String generateRandomNumber1(int length){
      StringBuffer randStr = new StringBuffer();
      for(int i=0; i<length; i++){
         int number = getRandomNumber1();
         char ch = NUM_LIST.charAt(number);
         randStr.append(ch);
      }
      return randStr.toString();
```

```
    }

    /**
     * This method generates random numbers
     * @return int
     */
    private int getRandomNumber() {
        int randomInt = 0;
        Random randomGenerator = new Random();
        randomInt = randomGenerator.nextInt(CHAR_LIST.length());
        if (randomInt - 1 == -1) {
            return randomInt;
        } else {
            return randomInt - 1;
        }
    }
    private int getRandomNumber1() {
        int randomInt = 0;
        Random randomGenerator = new Random();
        randomInt = randomGenerator.nextInt(NUM_LIST.length());
        if (randomInt - 1 == -1) {
            return randomInt;
        } else {
            return randomInt - 1;
        }
    }
     public static void main(String a[])throws Exception{
        KeyGenerator msr = new KeyGenerator();
        System.out.println(msr.generateRandomNumber1(1));
        System.out.println(msr.generateRandomString(5));

    }
}
```

Login

```
package com.auth;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import com.util.DbConnection;



/**
 * Servlet implementation class login
 */
@WebServlet("/login")
public class login extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public login() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		String email = request.getParameter("username");
		String password = request.getParameter("password");
		String id;
		String mob;
		HttpSession session = request.getSession();
		try {
			Connection con = DbConnection.getConnection();
			PreparedStatement ps = con.prepareStatement("select * from client where email=? and password=?");
			ps.setString(1, email);
			ps.setString(2, password);
			ResultSet rs = ps.executeQuery();
			if(rs.next())
			{


				id=rs.getString("id");
				mob=rs.getString("mobile");
				session.setAttribute("userid", id);
				session.setAttribute("email", email);

				PreparedStatement ps1 = con.prepareStatement("delete from auth where userid=?");
				ps1.setString(1, id);
				ps1.executeUpdate();
				new Thread().sleep(2000);
				response.sendRedirect("auth.jsp?a1");
			}
			else
			{
```

```
                                response.sendRedirect("index.jsp?invalid");
                        }

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

}
```

MD5 Code

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

package com.auth;

/**
 *
 * @author User6
 */
 import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class MD5 {




    public static void main(String text[]) throws Exception {
       String password = "pass@word1";


        System.out.println("Password: " + password + " in SHA512 is:");
        System.out.println("Tag: "+hashText(password));
        System.out.println(md5(password));
    }

   public static String convertByteToHex(byte data[])
   {
      StringBuffer hexData = new StringBuffer();
      for (int byteIndex = 0; byteIndex < data.length; byteIndex++)
```

```
            hexData.append(Integer.toString((data[byteIndex] & 0xff) + 0x100, 16).substring(1));

        return hexData.toString();
    }

    public static String hashText(String textToHash) throws Exception
    {
        final MessageDigest sha512 = MessageDigest.getInstance("SHA-512");
        sha512.update(textToHash.getBytes());

        return convertByteToHex(sha512.digest());
    }
     public static String md5(String input) throws NoSuchAlgorithmException {
        MessageDigest mDigest = MessageDigest.getInstance("SHA1");
        byte[] result = mDigest.digest(input.getBytes());
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < result.length; i++) {
            sb.append(Integer.toString((result[i] & 0xff) + 0x100, 16).substring(1));
        }

        return sb.toString();
    }
//}
   }
```

PGP Code

```
package com.auth;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;


import com.util.DbConnection;


/**
 * Servlet implementation class PGP
 */
```

```
@WebServlet("/PGP")
public class PGP extends HttpServlet {
	private static final long serialVersionUID = 1L;

  /**
   * @see HttpServlet#HttpServlet()
   */
  public PGP() {
    super();
    // TODO Auto-generated constructor stub
  }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		String userid = request.getParameter("id");
		String ticket = request.getParameter("ticket");

		try {
			Connection con = DbConnection.getConnection();
			GlobalFunction GF = new GlobalFunction();
			MD5 md = new MD5();

			PreparedStatement ps = con.prepareStatement("select * from auth where userid=?");
			ps.setString(1, userid);
			ResultSet rs = ps.executeQuery();
			if(rs.next())
			{
				String digest=rs.getString("digest");
				String tkt = rs.getString("ticket");
				//String mdigest = md.md5(ticket);
				String mdigest = md.md5(tkt);
```

```
                                if(digest.equals(mdigest))
                                {
                                        System.out.println("");
                                        System.out.println("Verifying your ticket digest for authorization.");
                                        KeyGenerator key = new KeyGenerator();
                                        String sign = "CSP"+key.generateRandomNumber1(3);
                                        PreparedStatement ps1 = con.prepareStatement("update auth set cspsign=? where userid=?");
                                        ps1.setString(1, sign);
                                        ps1.setString(2, userid);
                                        ps1.executeUpdate();
                                        new Thread().sleep(5000);
                                        response.sendRedirect("auth.jsp?a4");
                                }
                                else
                                {
                                        response.sendRedirect("index.jsp?hack");
                                }
                        }

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

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
                // TODO Auto-generated method stub
        }

}
```

Registration Code

```
package com.auth;
import java.io.IOException;
import java.sql.Connection;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;
import com.util.DbConnection;

/**
 * Servlet implementation class register
 */
@WebServlet("/register")
public class register extends HttpServlet {
	private static final long serialVersionUID = 1L;

   /**
    * @see HttpServlet#HttpServlet()
    */
   public register() {
      super();
      // TODO Auto-generated constructor stub
   }

	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

	}

	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		String fname = request.getParameter("fname");
		String lname = request.getParameter("lname");
		String dob = request.getParameter("dob");
```

```
                String gender = request.getParameter("gender");
                String city = request.getParameter("city");
                String email = request.getParameter("email");
                String mobile = request.getParameter("mobile");

                KeyGenerator mr = new KeyGenerator();
                String password = mr.generateRandomString(5);

                SendMailSSL send = new SendMailSSL();
                int flag=send.EmailSending(email, "Password", "Your Password : "+password);
                if(flag>0){
                try {
                        Connection con = DbConnection.getConnection();
                        PreparedStatement ps =con.prepareStatement("insert into client
(fname,lname,dob,gender,city,email,mobile,password) values(?,?,?,?,?,?,?,?)");
                        ps.setString(1, fname);
                        ps.setString(2, lname);
                        ps.setString(3, dob);
                        ps.setString(4, gender);
                        ps.setString(5, city);
                        ps.setString(6, email);
                        ps.setString(7, mobile);
                        ps.setString(8, password);
                        ps.executeUpdate();
                        response.sendRedirect("index.jsp?success");
                } catch (Exception e) {
                        e.printStackTrace();
                }
                }
                else
                {
                        response.sendRedirect("register.jsp?failed");
                }

        }

}
```

Request to Authentication Server

```
package com.auth;
import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import com.util.DbConnection;

/**
 * Servlet implementation class requestToAS
 */
@WebServlet("/requestToAS")
public class requestToAS extends HttpServlet {
	private static final long serialVersionUID = 1L;

   /**
    * @see HttpServlet#HttpServlet()
    */
   public requestToAS() {
      super();
      // TODO Auto-generated constructor stub
   }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
try {
        Connection con = DbConnection.getConnection();
        GlobalFunction GF = new GlobalFunction();
        String id = request.getParameter("id");
        String email = request.getParameter("email");
        String name[]=GF.getFullName(Integer.parseInt(id)).split(" ");
        AESAlgorithm aes = new AESAlgorithm();
        String enc = aes.encrypt("Hello "+GF.getFullName(Integer.parseInt(id))+" this is verification message", "AES");
        KeyGenerator rr = new KeyGenerator();
        String publickey = rr.generateRandomString(5);
        SendMailSSL send = new SendMailSSL();
        int flag = send.EmailSending(email, "Public Key", "Public Key = "+publickey);
        if (flag > 0) {
            PreparedStatement ps = con
                    .prepareStatement("insert into auth (fname,lname,email,userid,msg,publickey) values(?,?,?,?,?,?)");
            ps.setString(1, name[0]);
            ps.setString(2, name[1]);
            ps.setString(3, email);
            ps.setString(4, id);
            ps.setString(5, enc);
            ps.setString(6, publickey);
            int res = ps.executeUpdate();
            if (res > 0) {
                // System.out.println("Client sends request to Authentication Server for service....");
                System.out.println("");
                System.out.println("Authentication Server send encrypted message which is encrypted by private key to client for authentication....");
                System.out.println("Encrypted Message : " + enc);
                new Thread().sleep(5000);
                response.sendRedirect("auth.jsp?a2");
            }

        } else {
            response.sendRedirect("auth.jsp?failed");
```

```
                }
            } catch (Exception e) {

                e.printStackTrace();
            }

        }

        /**
         * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
         */
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {


            String pkey = request.getParameter("pkey");

            try {
                Connection con = DbConnection.getConnection();
                AESAlgorithm aes = new AESAlgorithm();

                PreparedStatement ps = con.prepareStatement("select * from auth where
publickey=?");
                ps.setString(1, pkey);
                ResultSet rs = ps.executeQuery();
                if(rs.next())
                {
                    PreparedStatement ps1 = con.prepareStatement("update auth set
asflag='ack'");
                    ps1.executeUpdate();
                    response.sendRedirect("auth.jsp?a22");
                }
                else
                {
                    response.sendRedirect("auth.jsp?a21");
                }
```

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

}
```

OTP Code

```
import java.util.Properties;

import javax.mail.Message;

import javax.mail.MessagingException;

import javax.mail.PasswordAuthentication;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeMessage;

 /*******  SSL:- (Secure Sockets Layer) ********/

public class SendMailSSL {


   Session session;

   String To;


public int EmailSending(String To, String Sub, String Msg) {

     System.out.println("----------To-----------"+To);

       int flag;

       try {

          Properties props = new Properties();

          props.put("mail.smtp.host", "smtp.gmail.com");

          props.put("mail.smtp.socketFactory.port", "465");

          props.put("mail.smtp.socketFactory.class",
```

```
        "javax.net.ssl.SSLSocketFactory");
props.put("mail.smtp.auth", "true");
props.put("mail.smtp.port", "465");

session = Session.getDefaultInstance(props,
        new javax.mail.Authenticator() {
           @Override
           protected PasswordAuthentication getPasswordAuthentication() {
              return new PasswordAuthentication("myproject3694@gmail.com","project@123");
           }
        });

String From = "subhash12784@gmail.com";


Message message = new MimeMessage(session);
// Set From: header field of the header.
message.setFrom(new InternetAddress(From));
// Set To: header field of the header.
message.setRecipients(Message.RecipientType.TO,
        InternetAddress.parse(To));
// Set Subject
message.setSubject(Sub);

// Now set the actual message
/* message.setText(Msg); */

// For set msg As HTML coding "Use Either .setText Or This Method to Send Msg"
```

```
            message.setContent(Msg, "text/html" );


            Transport.send(message);
            System.out.println("Sent message successfully....");


            flag = 1;
        } catch (MessagingException ex) {
            System.out.println("Exception "+ex);
            return -1;


        }
      return flag;
    }//SEND USER MAIL END


    public static void main(String[] args) {
                SendMailSSL s = new SendMailSSL();
        s.EmailSending("subhash12784@gmail.com", "Test", "Test");


    }
}//CLOSE MAIN CLASS
```

TGS Code

```
package com.auth;

import java.io.IOException;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
```

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

import com.util.DbConnection;

/**
 * Servlet implementation class TGS
 */
@WebServlet("/TGS")
public class TGS extends HttpServlet {
	private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public TGS() {
        super();
        // TODO Auto-generated constructor stub
    }

	/**
	 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
	 */
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		String userid = request.getParameter("id");

		try {
			Connection con = DbConnection.getConnection();
			GlobalFunction GF = new GlobalFunction();
			String fullname = GF.getFullName(Integer.parseInt(userid));
			String str[] = fullname.split(" ");
			KeyGenerator key = new KeyGenerator();
```

```
                    DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
            Date date = new Date();
                    String ticket = "TGT-"+dateFormat.format(date)+str[1]+str[0];
                    String clientsign = str[1]+str[0];
                    MD5 digest = new MD5();
                    String msgdigest=digest.md5(ticket);
                    PreparedStatement ps = con.prepareStatement("update auth set
ticket=?,digest=?,clientsign=? where userid=?");
                    ps.setString(1, ticket);
                    ps.setString(2, msgdigest);
                    ps.setString(3, clientsign);
                    ps.setString(4, userid);
                    ps.executeUpdate();
                    AuthenticationServer("TGT-"+dateFormat.format(date));
                    new Thread().sleep(2000);
                    response.sendRedirect("auth.jsp?a3");

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

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
            // TODO Auto-generated method stub
    }


    public void AuthenticationServer(String Ticket)throws Exception
    {
            System.out.println("");
            System.out.println("Authentication Server : Acknoledgement Received.");
            System.out.println("Authentication Server Send your credntial information to TGS");
            System.out.println("");
```

```
                new Thread().sleep(2000);
                System.out.println("TGS Generate Ticket and encrypted with session key.");
                System.out.println("Ticket : "+Ticket);
                new Thread().sleep(1000);
                System.out.println("Then send these tickets to Client and Real Server(CSP).");
                System.out.println("Calculate message digest for Integrity Checking.");
        }

}
```