IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. IX (Mar-Apr. 2014), PP 16-19
www.iosrjournals.org

Highly Confidential Security System

Cheruku Sandeep Kumar[1], Nandikonda Sindhuja[2,] Posham Lenin Reddy[3]
[1]*(Eee, Gitam University, India)*
[2]*(It, Griet,India)*
[3]*(Eee, Gitam University, India)*

Abstract : *In todays crazy busy life style it is not very uncommon for us all to be forgetful.We often fail to remember our passwords,mail ids,pancard numbers,passport details,study certificate numbers etc..,this kind of data is confidential and at present we store them manually (i.e mobiles,sticky notes) which is very easy to lose or even hacked.The "HIGHLY CONFIDENTIAL SECURITY SYSTEM" aims at developing a web application through which user can store his confidential data in a very secured way.*

Keywords: *Authentication,Details,Reports,Security,Web Applications.*

I. Introduction

Now a days its very common for every individual to have his/her personal data that is to be stored confidentialy.In existing system we are storing this data manually which doesnt provide minimum security for our data.This proposed web application will definitely overcome the demerits of our existing system.

The development of this new system contains the following activities, which try to automate the entire process keeping in the view of database integration approach with highly confidential security.

- This system maintains user data in encryption decryption format using algorithms.Here we are elliptic curve cryptography algorithm inorder to encrypt the user given data.
- This system maintains user's personal, address, and contact details.
- User friendliness is provided in the application with various controls provided by system rich user interface.
- This system makes the overall project management much easier and flexible.
- Various classes have been used for maintain the details of all the users and catalog.
- Authentication is provided for this application only registered users can access.
- Report generation features is provided using to generate different kind of reports.
- The system provides facilities to maintain bank account information.
- The system provides facilities to maintain Mails, password account information.
- The system provides facilities to maintain all education information marks memo, scaned copies information.
- The system provides facilities to maintain License, passport, insurances account information.
- The system provides facilities to maintain personal Files Information videos, images account information.
- System provides facility to online user registration.
- This system is providing accessibility control to data with respect to users.

- The coding of Elliptic Curve Cryptography is as fallows:

```
        import java.math.*;
    import java.util.*;

        public class ECC {

          // Parts of one ECC system.
          private EllipticCurve curve;
          private Point generator;
          private Point publicKey;
          private BigInteger privateKey;

          // We need a curve, a generator point (x,y) and a private key, nA, that will
```

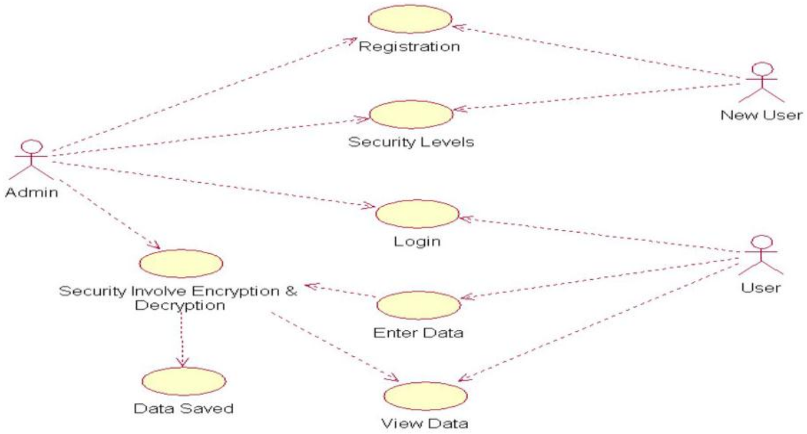
```
   // be used to generate the public key.
   public ECC(EllipticCurve c, BigInteger x, BigInteger y, BigInteger nA) {
curve = c;
              generator = new Point(curve, x, y);
              privateKey = nA;
              publicKey = generator.multiply(privateKey);
   }

   // Encryption.
   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 static void main(String[] args) {

              // Just use the book's curve and test.
              EllipticCurve myCurve = new EllipticCurve(new BigInteger("23"), new BigInteger("1"),
new BigInteger("1"));
              BigInteger x = new BigInteger("6");
              BigInteger y = new BigInteger("19");
              BigInteger nA = new BigInteger("10");
              ECC Alice = new ECC(myCurve, x, y, nA);

              // I have hard-coded my plaintext point.
              Point plain = new Point(myCurve, new BigInteger("3"), new BigInteger("13"));
              System.out.println("encrypting "+plain);

              // Encrypt and print.
Point[] cipher = Alice.encrypt(plain);
              System.out.println("cipher first part "+cipher[0]);
              System.out.println("cipher second part "+cipher[1]);
```

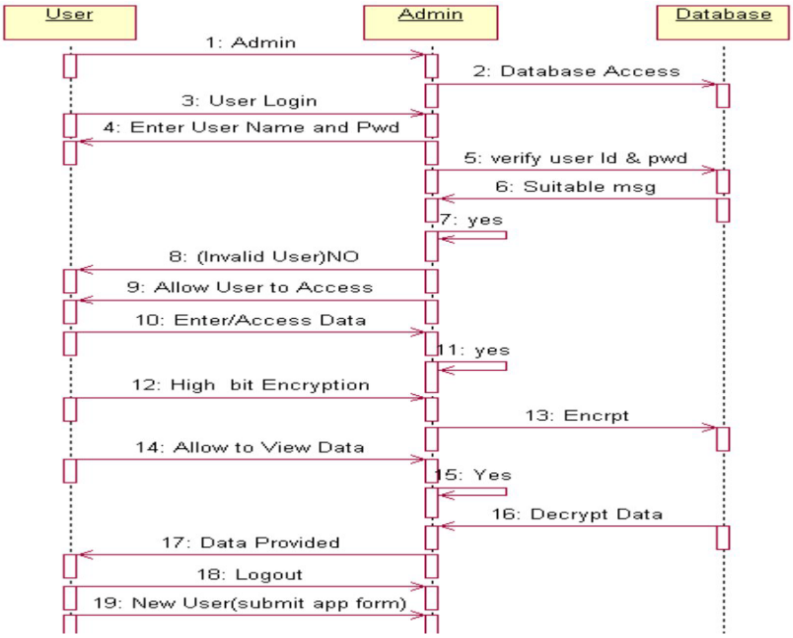
```
                // Decrypt and verify.
                Point recover = Alice.decrypt(cipher);
                System.out.println("recovered "+recover);

     }
  }
```

SOFTWARE AND HARDWARE REQUIREMENTS

Operating System	Windows XP/2007 or Linux
User Interface	HTML, CSS
Client-side Script	JavaScript
Programming Language	Java
Framework	struts 1.x, Hibernate 3.0
IDE/Workbench	My Eclipse 8.6
Database	Oracle 10g
Server Deployment	Tomcat 6.0/7.0
Processor	Core 2 Duo
Hard Disk	160GB
RAM	1GB or more

- Since it is a web application any number of users can access his/her account at any time from any place





IG1.1:SEQUENCE DIAGRAM

The use case diagram in fig1.0 describes the following :
ADMIN:He can allow registration , he can check the security levels,he can validate the login ,he involves in security provision.
USER:He can login,he can enter the data ,he can view the data, he can update the data.
NEWUSER:He can register,he can view the security levels.

The sequence diagram of fig1.1 describes the flow of the work done by the web app.

II. Conclusion

ADVANTAGES:

- Provides the best security for our data.
  We can store our data like:
  All mailids, passwords
  All bank account no
  Insurance policy No
  PAN NO
  Driving License No
  All education certificate Numbers
  Some highly value scan copy
  Some confidential photo and music, videos
  - We can update the details,delete details.
  - On request of the user a copy of this details is provided(if any case of death of the user we can provide this details to his family)

LIMITATIONS:

- If he types password wrong for more than 3 times his account will be blocked and he need to give the proper reason through mail with the unique id that will be provided to him during registration and the password will be provided to him as a response.
- There is the limitation of video files that user has secured.

EXTENSIONS:

- This project can be extended in future so that we can directly do the bank transactions using this data provided in the web site.

References:

[1] Cryptography and Network Security,Principles and Practices,Fourth Edition,William Stalings,PEARSON Education.
[2] The Complete Reference Java2,Third Edition,Patrick Naughton,Herbert Schildt,TATA McGraw-HILL EDITON
[3] Web Component Development With Servlet And JSP Technologies,SUN EDUCATIONAL SERVICES,SUN MICROSYSTEMS
[4] Web Programming,Building Internet Applications,Second Edition,Chris Bates,Sheffield Hallam University,WILEY INDIA
[5] w3schools.com