Final Project Report

Elliptic Curve cryptography based mutual authentication scheme for session initiation protocol

Course Code: ICSI526 Spring 2018, Cryptography

Course Instructor: **Dr. Pradeep K. Atrey**

Date of Submission: May 9th, 2018

By,

Name	Student Id #	Contact Information
B. Harisankar Prusty	001351919	hprusty@albany.edu
Shivika Malik	001346591	smalik2@albany.edu

Department of Computer Science

The University at Albany

State University of New York

ABSTRACT

The session initiation protocol (SIP) is the most widely used signalling protocol for controlling communication on the internet, establishing, maintaining, and terminating the sessions. The services that are enabled by SIP are equally applicable in the world of wireless communication. Recently, Tsai proposed an efficient nonce-based authentication scheme for SIP. In this paper, we do a cryptanalysis of Tsai's scheme and show that Tsai's scheme is vulnerable to the password guessing attack and stolen-verifier attack. Furthermore, Tsai's scheme does not provide known-key secrecy and perfect forward secrecy. We also propose a novel and secure mutual authentication scheme based on elliptic curve discrete logarithm problem for SIP which is immune to the presented attacks. Session Initiation Protocol (SIP) as the controlling protocol has attracted much attention. SIP is one of the most widely used for securing and controlling communication over the Internet.

1.Introduction:

Multimedia service is one of the most important application classes of today's and of the future's networks. It's very important to try our best to improve the multimedia service. As important protocols support multimedia services, the session initiation protocol (SIP) has been widely studied since its emergence. Security has the highest importance in data communication world. When It comes to authentication, User should have the previlage to use a secure media to connect with any of the server.

1.1.Background:

SIP is widely used signaling protocol for controlling the communication on the internet, maintaining, establishing as well as terminating the sessions. It is protocol of the application layer which can create, modify and terminate sessions with the participants. It supports multimedia sessions/services on both wired and wireless networks like multimedia distribution, internet telephone calls. Session member can communicate via a mesh of unicast or multicast relations. We are implementing Cryptanalysis of Tsai’s scheme and show that the scheme is vulnerable to password guessing attack and stolen verifier attack. And also implement a secure mutual authentication which is based on Elliptical Curve Discrete Logarithm problem for SIP which is immune to the attacks.

Authentication is a crucial process when a remote user wants to get services from a corresponding sever. Most environment communications of SIP are unsafe which naturally raises the issue of providing security protection for communicating participants. Therefore, we are trying to design a robust and efficient mutual authentication which is meaningful and interesting.

Elliptical Curve Cryptography

1. An elliptic curve E over Fp is set of all solutions (x, y) □Fp * Fp to the equation E: y2 = x3 +ax + b (mod p)
2. p>3 be a large prime number and point P be of a prime order where nP = O and P ≠ O, where O is a special point at infinity serving as an identity element.
3. E is an(additively written) abelian group with the point O serving as its identity element.
4. ECC is defined as Ep(a, b) : y2 = x3 + ax +b (mod p) over prime finite field Fp, where b, a □ Fp and 4a3 + 27b2 □ 0 (mod p)
5. Given the points A, B over Ep (a, b), the computational discrete logarithm problem is to decide m□ Fp* from B = mA
6. mP, nP are the given points over Ep(a, b), ECCDHP is to compute mnP.
7. Given the points mP, P over Ep(a, b), the inverse computational Diffie-Hellman problem is to find m-1P

1.2.Scope:

The scope of this project is to design a robust and efficient mutual authentication scheme for session initiation protocol and its security analysis. Scope includes authenticating the user with server in a secured manner with proper use of Elliptical cryptographic protocol.

1. It can be used in any of the field to hide the user specific data and which contais personal information regarding the user.

1.3.LIMITATION:

It was difficult to implement session initiation protocol using java, because there were lot of algorithm which cannot be implemented with out any library.

2. ALGORITHM AND IMPLEMENTATION

The user in the authentication scheme for SIP is allowed to choose his password freely. Generally speaking, the user would want to choose a password that can be easily remembered for his or her convenience. However, these easy-to-remember passwords are potentially vulnerable to the password-guessing attacks, in which an adversary can try to guess the client's password and then verify his guesses . The password-guessing attack can be classified into on-line password-guessing attack and off-line password-guessing attack. In the on-line password-guessing attack, the adversary tries to use guessed passwords iteratively to pass the verification of the server in an on-line manner. Whereas in the off-line password attack, the adversary intercepts some password-related messages exchanged between the user and the server and then iteratively guesses the client's password and verifies whether his guess is correct or not in an off-line manner. On-line password-guessing attacks can be easily thwarted by limiting the number of continuous login attempts within a short period. However, the off-line password-guessing attacks cannot easily be found because there is no need for the server to participate in the verification.

2.1. TECHNIQUES AND METHODOLOGIES

Elliptical Curve Cryptography

1. An elliptic curve E over Fp is set of all solutions (x, y) $\in$ Fp * Fp to the equation E: $y^2 = x^3 + ax + b \pmod p$

2. p>3 be a large prime number and point P be of a prime order where nP = O and P $\neq$ O, where O is a special point at infinity serving as an identity element.

3. E is an(additively written) abelian group with the point O serving as its identity element.

4. ECC is defined as Ep(a, b) : $y^2 = x^3 + ax + b \pmod p$ over prime finite field Fp, where b, a $\in$ Fp and $4a^3 + 27b^2 \neq 0 \pmod p$

5. Given the points A, B over Ep (a, b), the computational discrete logarithm problem is to decide m $\in$ Fp* from B = mA

6. mP, nP are the given points over Ep(a, b), ECCDHP is to compute mnP.

7. Given the points mP, P over Ep(a, b), the inverse computational Diffie-Hellman problem is to find m-1P

2.2 IMPLEMENTATION

Code is written in JAVA. We are going to write separate functions into a separate file as needed as mentioned below. We are using some of the built in libraries in java to perform operations on ECC. We are going to proceed the project in following steps:

Phase: Registration

When the client user wants to register and become a legal user, then the client user and the Server execute the steps over a secure channel.

1. Client User -> Server : {username, password}

User submits username and password to the Server. Server computes 2 secret values, that is, HPWD and HKs by using hash of client user's -

- username
- password, PW
- Ks: Shared Secret Key

HPW = h(username || PWD) and HKs = h(username || Ks)

2. Server computes the password verifier

VPW = HPW $\oplus$ HKs for User

3. Server stores the User's username and the verified password in the user account database.

Phase: Authentication

If a legal user wants to login into a Server, then the user is supposed to type the username and the password to authenticate.

1. Client User to Server: REQUEST (username, PW)

User generates r1 and computes HPW and then computes $R_1 = (HPW.r_1)P$ and sends it with a request message to Server.

2. Server to User: CHALLENGE (realm, R2, h1)

Server extracts HPW from VPW by computing HKs. Then, Server computes $R'_1 = HPW^{-1} R_1 = (HPW^{-1}.HPW.r_1)P = r_1P$ HPW^{-1} is computed by Extended Euclidean Algorithm. S generates r_2 and computes $R_2 = r_2P$, $SKs = r_2R'_1 = r_2r_1P = r_1r_2P$ and $h_1 = h(SKs \| R_2)$

And $SKs = r_2r_1P = r_1r_2P$

SKs holds due to commutative property of elliptic curve group.

Finally, Server sends a challenge message to the User.

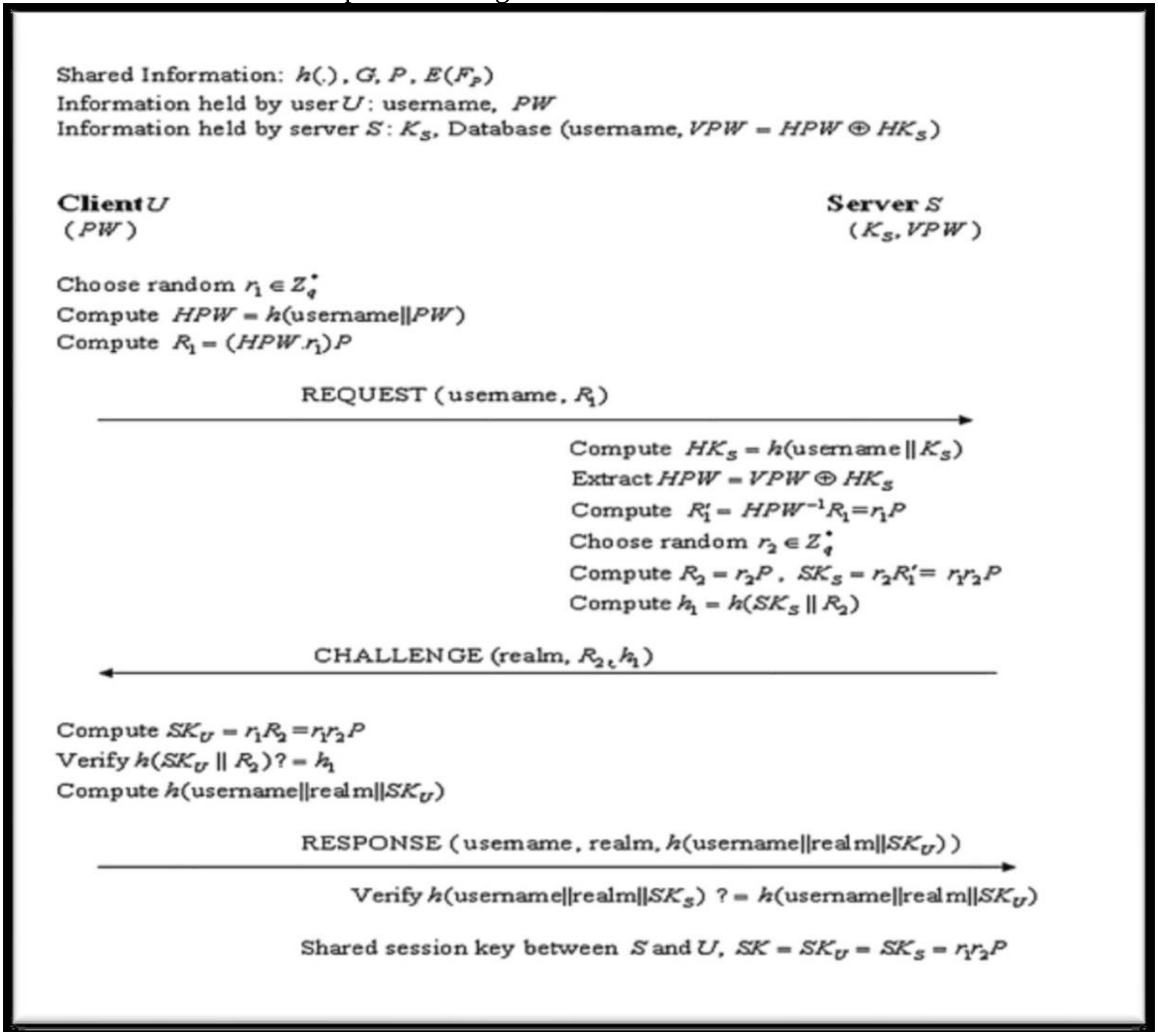
3. User → Server: RESPONSE (username, realm, h(username||realm||SKU))

When the User receives the CHALLENGE message, user computes SKU = r_1R_2 = r_1r_2P and checks if h(SKU || R_2) = h_1

If **TRUE**, user authenticated the server and computes a message authentication code:

h(username || realm || SKU)

and then the user sends a response message to server:



Shared Information: $h(.)$, G, P, $E(F_p)$

Information held by user U: username, PW

Information held by server S: K_S, Database (username, $VPW = HPW \oplus HK_S$)

Client U (PW)

Server S (K_S, VPW)

Choose random $r_1 \in Z_q^*$

Compute $HPW = h(\text{username}||PW)$

Compute $R_1 = (HPW.r_1)P$

REQUEST (usemame, R_1)

Compute $HK_S = h(\text{username}||K_S)$

Extract $HPW = VPW \oplus HK_S$

Compute $R_1' = HPW^{-1}R_1 = r_1P$

Choose random $r_2 \in Z_q^*$

Compute $R_2 = r_2P$, $SK_S = r_2R_1' = r_1r_2P$

Compute $h_1 = h(SK_S || R_2)$

CHALLENGE (realm, R_2, h_1)

Compute $SK_U = r_1R_2 = r_1r_2P$

Verify $h(SK_U || R_2) ?= h_1$

Compute $h(\text{username}||\text{realm}||SK_U)$

RESPONSE (usemame, realm, $h(\text{username}||\text{realm}||SK_U)$)

Verify $h(\text{username}||\text{realm}||SK_S)$?= $h(\text{username}||\text{realm}||SK_U)$

Shared session key between S and U, $SK = SK_U = SK_S = r_1r_2P$

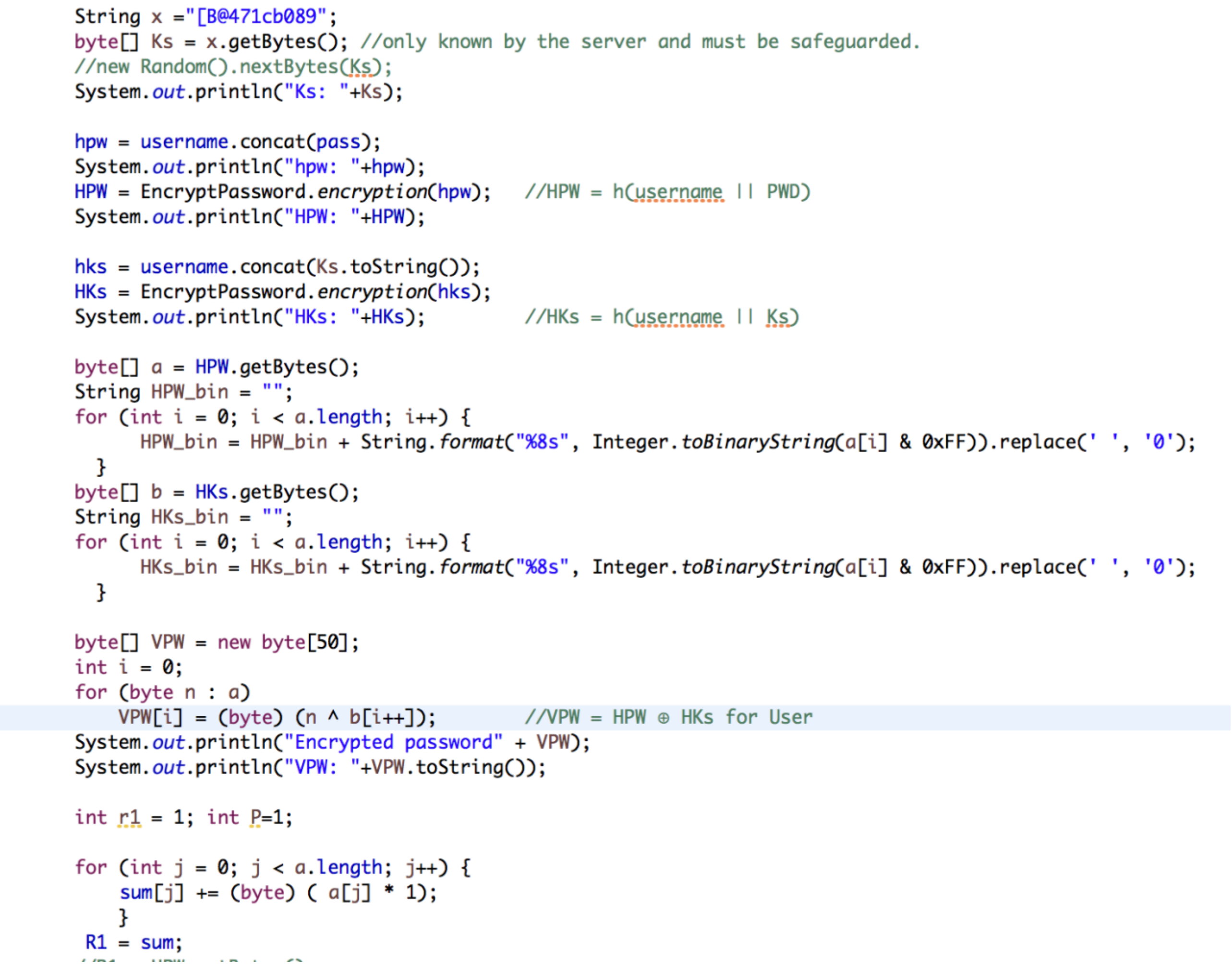
Figure 1:Algorithm used for implementation

RESPONSE (username, realm, h(username || realm || SKU)) Else user rejects the server challenge message.
4. After receiving the response message, Server computes h(username||realm||SKs) and verifies it with the received response h(username||realm||SKU).

If **TRUE**, Server authenticates user and accepts the User's login request Else, Server rejects the User response message.
SK = SKU = SKs = r_1r_2P is used as a shared session key after mutual authentication between User and the Server.

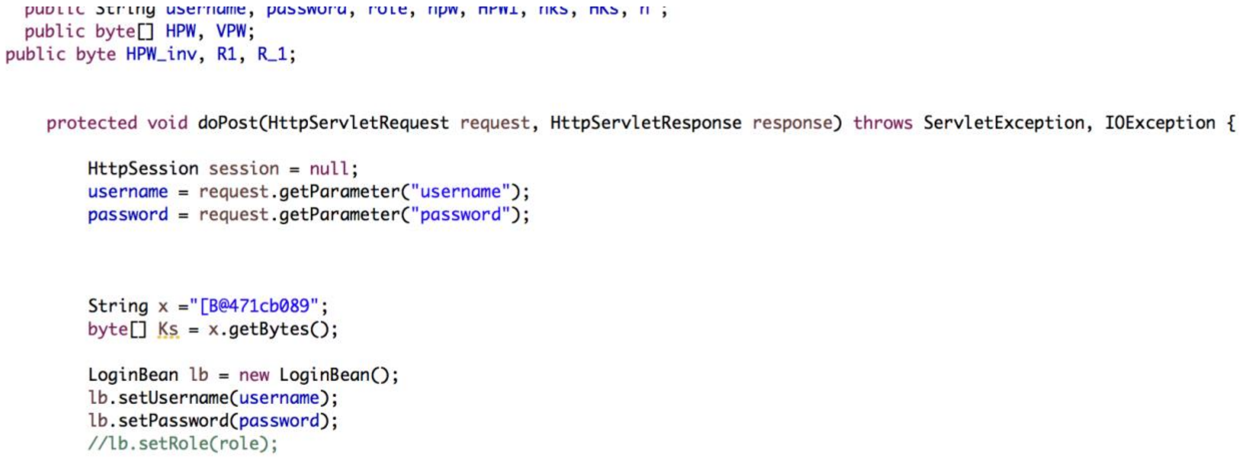
2.3 ScreenShots

2.3.1 Registration Phase:

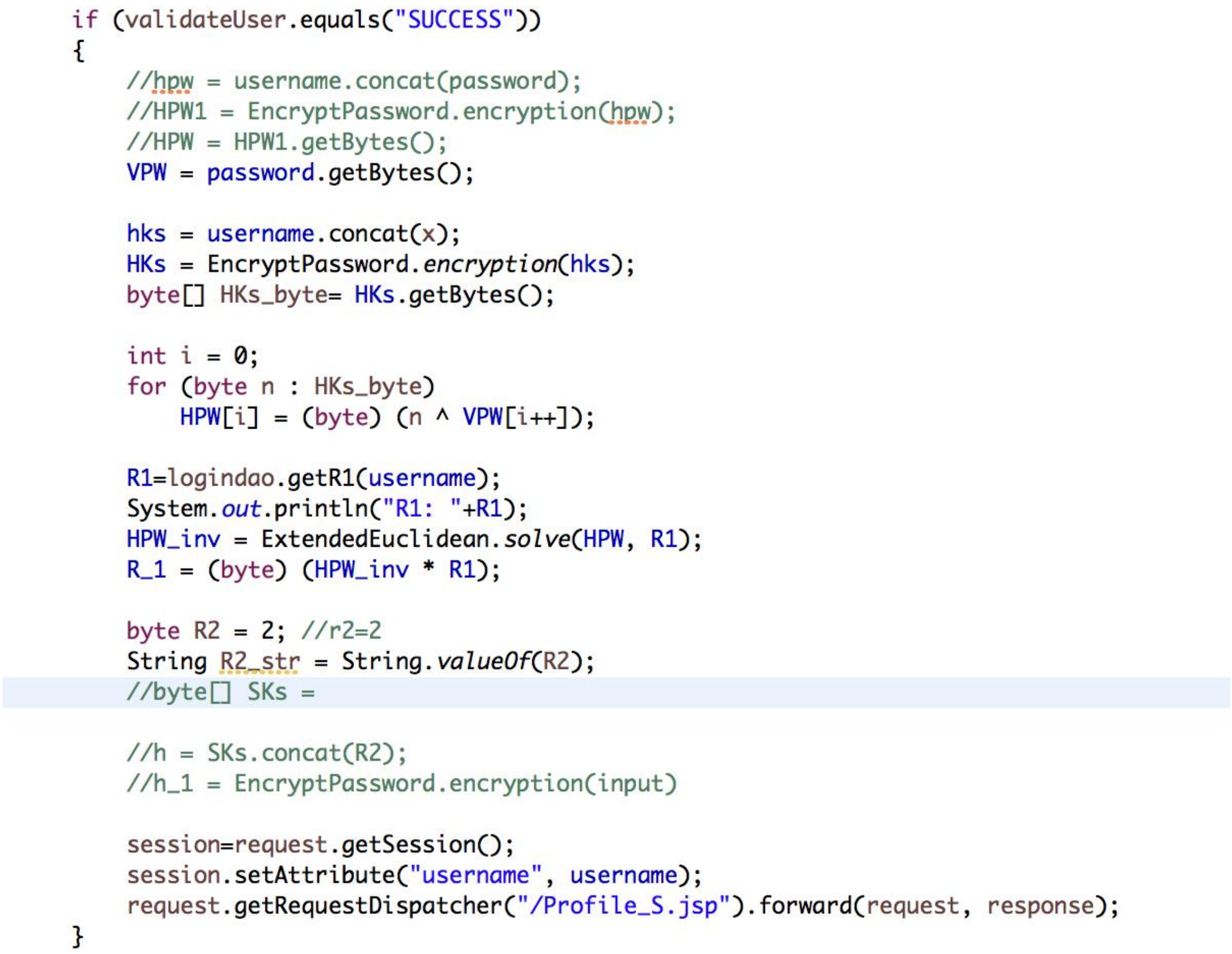


```
String x ="[B@471cb089";
byte[] Ks = x.getBytes(); //only known by the server and must be safeguarded.
//new Random().nextBytes(Ks);
System.out.println("Ks: "+Ks);

hpw = username.concat(pass);
System.out.println("hpw: "+hpw);
HPW = EncryptPassword.encryption(hpw);   //HPW = h(username || PWD)
System.out.println("HPW: "+HPW);

hks = username.concat(Ks.toString());
HKs = EncryptPassword.encryption(hks);
System.out.println("HKs: "+HKs);         //HKs = h(username || Ks)

byte[] a = HPW.getBytes();
String HPW_bin = "";
for (int i = 0; i < a.length; i++) {
      HPW_bin = HPW_bin + String.format("%8s", Integer.toBinaryString(a[i] & 0xFF)).replace(' ', '0');
  }
byte[] b = HKs.getBytes();
String HKs_bin = "";
for (int i = 0; i < a.length; i++) {
      HKs_bin = HKs_bin + String.format("%8s", Integer.toBinaryString(a[i] & 0xFF)).replace(' ', '0');
  }

byte[] VPW = new byte[50];
int i = 0;
for (byte n : a)
    VPW[i] = (byte) (n ^ b[i++]);        //VPW = HPW ⊕ HKs for User
System.out.println("Encrypted password" + VPW);
System.out.println("VPW: "+VPW.toString());

int r1 = 1; int P=1;

for (int j = 0; j < a.length; j++) {
    sum[j] += (byte) ( a[j] * 1);
    }
 R1 = sum;
```

2.3.2 AUTHENTICATION PHASE:

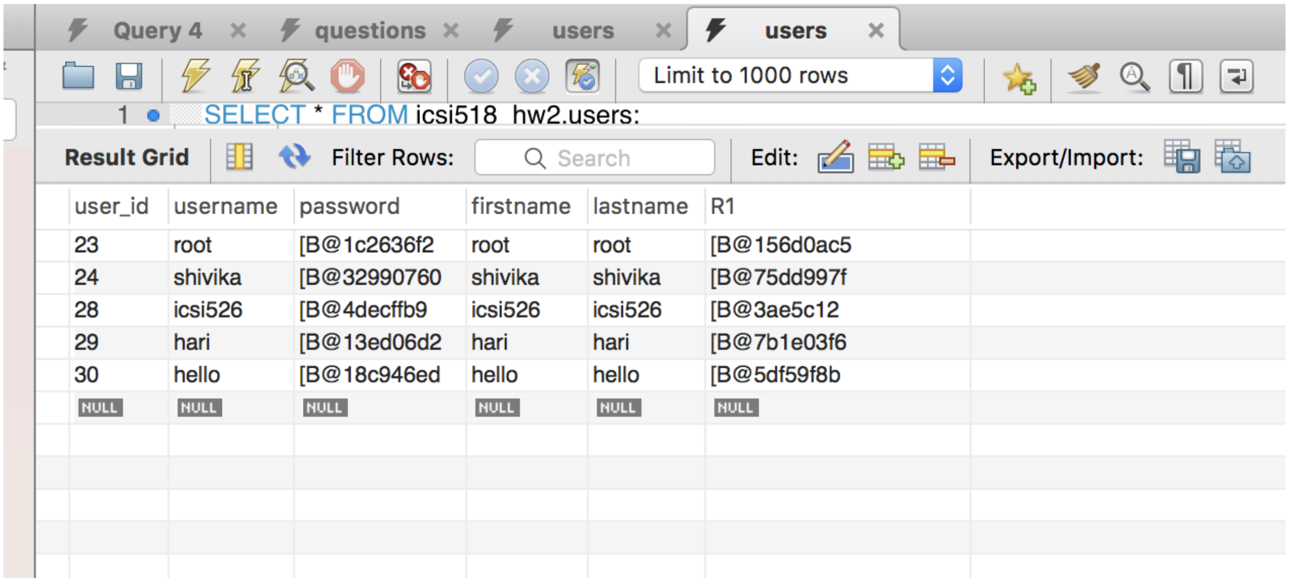


```
  public String username, password, role, hpw, HPW1, hks, HKs, h ;
  public byte[] HPW, VPW;
public byte HPW_inv, R1, R_1;


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

        HttpSession session = null;
        username = request.getParameter("username");
        password = request.getParameter("password");



        String x ="[B@471cb089";
        byte[] Ks = x.getBytes();

        LoginBean lb = new LoginBean();
        lb.setUsername(username);
        lb.setPassword(password);
        //lb.setRole(role);
```



```
if (validateUser.equals("SUCCESS"))
{
    //hpw = username.concat(password);
    //HPW1 = EncryptPassword.encryption(hpw);
    //HPW = HPW1.getBytes();
    VPW = password.getBytes();

    hks = username.concat(x);
    HKs = EncryptPassword.encryption(hks);
    byte[] HKs_byte= HKs.getBytes();

    int i = 0;
    for (byte n : HKs_byte)
        HPW[i] = (byte) (n ^ VPW[i++]);

    R1=logindao.getR1(username);
    System.out.println("R1: "+R1);
    HPW_inv = ExtendedEuclidean.solve(HPW, R1);
    R_1 = (byte) (HPW_inv * R1);

    byte R2 = 2; //r2=2
    String R2_str = String.valueOf(R2);
    //byte[] SKs =

    //h = SKs.concat(R2);
    //h_1 = EncryptPassword.encryption(input)

    session=request.getSession();
    session.setAttribute("username", username);
    request.getRequestDispatcher("/Profile_S.jsp").forward(request, response);
}
```

2.3.3 Database:



```
1 SELECT * FROM icsi518_hw2.users;
```

user_id	username	password	firstname	lastname	R1
23	root	[B@1c2636f2	root	root	[B@156d0ac5
24	shivika	[B@32990760	shivika	shivika	[B@75dd997f
28	icsi526	[B@4decffb9	icsi526	icsi526	[B@3ae5c12
29	hari	[B@13ed06d2	hari	hari	[B@7b1e03f6
30	hello	[B@18c946ed	hello	hello	[B@5df59f8b
NULL	NULL	NULL	NULL	NULL	NULL

3.Novel Contribution

Server authenticates on the security analysis of the project , which contains the below different attack listed below.

1.Password guessing attack

2. Man- in- middle- attack

3.Mutual Authentication Attacks

Below is the listed comparison contributed in the analysis of algorithm complexity

Table 1. Comparisons between our protocol and the protocol of Arshad *et al.*

Computational Cost Analysis	**Protocol of Arshad *et al.***	**Our protocol**
Computational cost (client)	$2TG_{mul} + 1T_{mul} + 3T_h$	$3TG_{mul} + 3T_h$
Computational cost (server)	$3TG_{mul} + 1T_{inv} + 3T_h + 1T_{XOR}$	$3TG_{mul} + 3T_h + 1T_{XOR}$

Computational Cost Analysis	**Protocol of Arshad *et al.***	**Our protocol**
Resist off-line password-guessing attack	No	Yes

4.EFFORT CONTRIBUTION

4.1 TEAM EFFORT:

Tasks Planned	Work Distribution	Status
Study of the selected Project paper.	Studied the research paper to get basic understanding.	Completed
Reading and analysing the IEEE paper and found out the appropriate java	Installed java on our laptops and executed few examples.	Completed
Research regarding implementation of ECC and search related to elliptic curve	Slides were covered and search related to 3 type of elliptic curve .	Completed
Theoretical Study on Session initiation protocol and its implementation	IEEE paper and related topics were researched and worked on it.	Completed
Study of code in online as how to implement SIP as both were new to SIP.	Pre-requisites were identified and coding implementation has started	Completed
Implemented the algorithm using java(jsp in front end and mvc architecture in backend)	Implemented using java, with proper results captured in the backend	Completed

4.2 Individual Contribution

Tasks Planned	Work Distribution	Project Member
Frontend Design	JSP pages and MVC	Shivika Malik
Backend design	Spring architecture	Harisankar Prusty
Database design	Worked on SQL WorkBench	Both
Study regarding elliptic curve	Complete paper study with lot of research regarding implementation	Both
Final Report creation	Report consist of our final outcomes and research regarding this project	Harisankar Prusty
Final Presentation	Worked on embedding of data after block shifting method	Shivika Malik

5.Conclusion

In this paper, we have shown that the authentication scheme by Arshad *et al.* for session initiation protocol is vulnerable to the password-guessing attack. In order to resolve these security problems, we have proposed a novel and secure mutual authentication scheme based on the ECC. The proposed authentication scheme not only resists these attacks but also provides greater security and efficiency. Hence, our proposed authentication scheme is much better than any other previously proposed related schemes.

References:

1. Arshad R, Ikram N. Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. *Multimedia Tools and Applications*. DOI: 10.1007/s11042-011-0787-0, 2011
2. Tsai J. Efficient nonce-based authentication scheme for session initiation protocol. *Int J Netw Secur* 2009; **8**(3): 312–316.
3. Yoon E, Yoo K, Kim C. A secure and efficient SIP authentication scheme for converged VoIP networks. *Computer Communications* 2010; **33**: 1674–1681.
4. Wu L, Zhang Y, Wang F. A new provably secure authentication and key agreement protocol for SIP using ECC. *Computer Standards and Interfaces* 2009; **31**(2): 286–291.
5. Code Reference:
   - http://ieeexplore.ieee.org/abstract/document/7582711/
   - https://link.springer.com/article/10.1007/s11042-011-0787-0#Sec18
   - https://github.com/DhruvDixitDD/ElGamal-based-Elliptic-Curve-Cryptography/blob/master/ElgamalEllipticCurve.py
   - https://stackoverflow.com/questions/13698624/sip-java-api-library