CS6740 NETWORK SECURITY

PS-4: SECURE INSTANT MESSAGINE SYSTEM

INSTALLATION/USAGE INSTRUCTIONS

Team 5
Arpit Mehta
Ananthanarayanan Balachandran

Architecture:

In our Secure Instant Messaging System, we use client-server architecture. At the client end, a user is required to remember the password which is included in the ClientNameList file. The Server maintains the RSA key pairs for all users. And in order to have augmented strong password property, the user's private key is encrypted with a function derived from the user's username & password. The server also maintains a weak secret that's derived from the user's password (different from the ones that are used to encrypt the users' private keys.

Assumptions:

1) Client needs to remember only his/her username & password.
2) Server generates and maintains RSA key pairs for all registered users.
3) Pre-registration of user's id done by running SIMHelper.java. This application basically generates the required resource files for the server and all the users for successful communication.
4) Augmented Strong Password Security is obtained by encrypting user's RSA private encrypted with W' = hash(username | password).
5) Server also maintains W = hash(password)
6) Java SE 6.0 used

Instructions to launch application:

Before launching the SIM application, the SIMHelper.java needs to be executed in order to configure the usernames and password of the users for the server. To launch the SIMHelper program following commands must be executed from the command prompt:

```
$: javac SIMHelper.java
$: java SIMHelper
```

Once the SIMHelper application is running, 4 users can be configured for the SIM system. An example snippet of execution of SIMHelper is:

```
$: username: user1
$: password: user1
$: username: user2
$: password: user2
$: username: user3
$: password: user3
$: username: user4
$: password: user4
```

In case the Server and clients are running on different machines, please ensure that the following:

1) The following files are needed by the server part of the application:
    a. SIMHelper.java
    b. SIMServer.java
    c. SIMServarCryptoHelper.java
2) The following files are needed by the server part of the application:
    a. SIMClient.java
    b. SIMClientHelper.java
    c. SIMClientCryptoHelper.java
3) Please ensure that the 'serverpublickey' file (which is generated when SIMHelper application is executed in the machine in which the server application is to run) is copied into the current directory of the SIMClient (or machine in which the client application is to run).

To launch the Server, the port number must be provided. Following is an example that illustrates how to launch the SIMServer application:

```
$: javac SIMServer.java
$: java SIMServer 2020
```

Following is the output when the server is initialized:

```
Server Initialized, listening to the port: 2020
```

Now the server is listening for incoming messages on the port 2020

To launch the Client, the server IP address and the server port number must be provided. Following is an example that illustrates how to launch the SIMClient application:

```
$: javac SIMClient.java
```

If the server application is running on local host, then the following command maybe used:

```
$: java SIMClient localhost 2020
```

If the server application is not running on local host, then the following command maybe used:

```
$: java SIMClient <ip> 2020
```

Once the client application is successfully initialized, the following User Messages would appear:

```
Connected To Server successfully!
Hello! Please use the following commands:
Log In: > login <username> <password>
List Of Online Users: > list
Send Message To User: > send <username> <message>
Log Out: > logout
```

To Login to the server, the following command must be entered in the client application console:

```
> Login user1 user1
```

Upon successful Login, the client console would display the following message:

```
User Login Successful!
```

To request the List of current online users, enter the following command:

```
> list
```

An example response to the list command from the user would be displayed like this:

```
user2 address: 127.0.0.1:61145
user1 address: 127.0.0.1:64729
```

NOTE: Before sending commands to other user, all running client applications must request the server for the list of online users using “list” command. This is needed for updating the local client HashMaps.

To Send message to another user, the user needs to enter the following commands on the client console:

```
> send <username> <message>
```

For example, id “user1” is sending a message “Hi” to “user2”, the following command must be used

```
> send user2 Hi
```

Once, the message is sent successfully, following would be displayed on the “user1” console

```
Message sent to user2:    hi
```

And the following would be displayed on “user2” console:

```
Message from user1:   hi
```

To Logout, the user ned to type the following command in the client console:

```
> logout
```

Upon successful logout, the following message would be displayed on the client console:

```
Successfully Logout!
```