Information Security Course Project

Feng Chaoyi, Xiao Linlu

14212010002@fudan.edu.cn, 14212010035@fudan.edu.cn

Room 405, Software Building

April 28, 2015

1 Abstract

1.1 Background

Suppose you want to communicate with your friends in a secret and secure way. However, the network environment is insecure, because some evil guys may eavesdrop or fake messages to corrupt your communication. To protect your communication from these possible threats, you want to develop an secure instant messaging (IM) application using knowledge you have learned in this course.

1.2 Project Motivation

You should use what you have learned in class to ensure security of this instant communication application. After this project, you should be familiar with security techniques such as different methods of encryption and decryption (symmetric and asymmetric), MAC (message authentication code) and verification to encrypted messages.

1.3 Development Environment

You should develop based on Java programming language. Java 6 or higher version is recommanded.
We don't have high-level requirements on GUI, you can use web pages or simple Swing GUI, so focus on the functionalities :)

2 Details

2.1 Functionality

Your program should be capable of following functionalities:

- **Registration** User should be able to register with his/her unique public identity (e.g. email address) to get an account (e.g. a public key) and key for the account (e.g. the corresponding private key) from an authority.
- **Friending** User A should be able to get user B's account by sending B's public identity to the authority. Then A is able to send a friend request to B's account. B, having received this request, should be able to approve or deny the request. If the request is approved, both users will become friends and should be able to exchange messages (A user should not be able to communicate to another unknown user).
- **Messaging** User should be able to send an encrypted message to his/her friend (which can't be eavesdropped or modified by evil guys).
- **(Optional bonus) File sending** User A should be able to send a simple file to user B while communicating, when the file can't be readable to the middle man, and if the file is maliciously modified an error would be reported.

2.2 What to implement

You should implement this program as following:

- Your program should be made up of two parts: `server` and `client`. There should be only one server, but there might be several clients running at the same time. Server is responsible for registration and account requerying (i.e., responding with some user A's account when queried with A's public identity). Client is responsible for sending friend requests, sending and receiving messages.
- Servers and clients do not necessarily need be on different computers. In this project, they could be different processes running on the same computer.
- Server and clients communicates using `sockets`. You may find more information on this topic at `JDK` documentation.

- Your program should be capable of protecting your messages, so it must encrypt the content of the message. The encryption algorithm is not specified.
- For efficiency, you should use `session key` for encryption.
- Some bad guys may want to modify your message, make sure you employ some method to guarantee the integrity of the messages you send.
- You should use Java libraries to do low-level work such as encryption and digital signature signing. **Don't implement your own encryption library! That would waste your work.**

2.3 What to hand in

- A detailed document explaining your design
- The source code
- The executable program and a manual to it

You should upload your work to folder `WORK_UPLOAD/Project1/` under corresponding course folder at FTP site: `ftp://10.132.141.33/`. All the above files should be compressed into a `.zip` file named after your student number and name, e.g., `12302010001-FullNameInEnglish.zip`.

2.4 Deadline

The deadline will be **2015/5/31 23:59:59 GMT+8:00**. Start early.

3 Grading

Your project will be graded according to following factors:

Design	**30%**
Reliable key distribution	5
Reliable account querying	5
Secure key exchange	5
Secure message sending	5
Integrity of message	5
Efficiency	5
Correctness of the executable program	**50%**

Registration and key distribution	10
Friending and authentication	10
Key generation and exchange	10
Message encryption and decryption	10
Message integrity	10
Quality of source code	**10%**
Bonus	**10%**
File sending	10

Warning: DO IT YOURSELF! Or your grade would be 0 directly.

4 Q&A

If you have any problem regarding this project, please contact Feng Chaoyi at `14212010002@fudan.edu.cn`, or Xiao Linlu at `14212010035@fudan.edu.cn`.