Crypto-IM

Introduction

Brief Overview

The Crypto-IM project is a cryptographically secure instant messaging application developed in Java. It allows users to choose what security constraints they desire for their chat, sign in, and then begin messaging a second party. The application implements two-way communication between two parties, the server and the client.

Dependencies

In order to run *Crypto-IM* you will need to install the following:

- git (optional)
- java 9

Crypto-IM has no further dependencies!

Build/Installation

To install *Crypto-IM* you can either hit the **Clone or download** button at the top of this repository, or use git to clone:

```
$ git clone https://github.com/ejrbuss/Crypto-IM
```

To build the project you can either import the project into the Eclipse IDE or run the following commands on the command line. Ensure that your current working directory is the root of this repository.

To compile the project run:

```
$ javac -d bin src/com/local/se360/*.java
```

Running Crypt-IM

Once compiled, the program can be started from the root directory of the repository. Run the server program with:

```
$ java -cp ./bin com.local.se360.Server
```

Run the client program with:

```
$ java -cp ./bin com.local.se360.Client
```

Design Decisions

Java & Javax Libraries

- **javafx** UI library
- **java.net** Socket programming
- **java.security** Signing and signature verification, user authentication
- **javax.crypto** Symmetric encryption/decryption

Cryptography Approaches

Confidentiality: Symmetric Encryption/Decryption

128 bit AES was used for Symmetric Key Encryption/Decryption.

Integrity: Signing and Signature Verification

256 bit RSA encryption with MD5 hashing was used for signature creation and verification.

Authentication: User Authentication

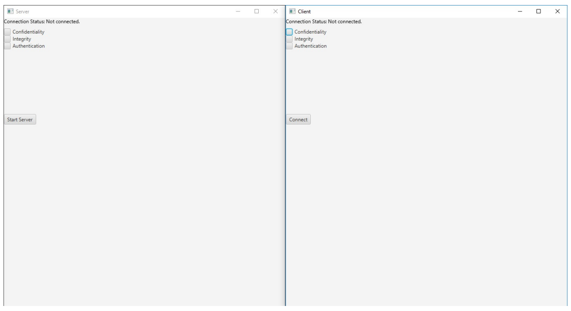
The SHA-256 hasing algorithm was used to hash user passwords which were then stored in a file for use when authenticating.

Limitations

Because *Crypto-IM* conducts an unecrypted diffie hellman key exchange it is vulnerable to a man in the middle attack during the client and server's first to handshake packets. For this to be resolved a certificate authority would need to correspond with the client and server or the public keys of both parties would have to be previously distributed.

User Walkthrough

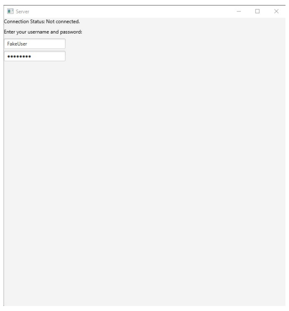
1. Open server and client application



2. Select desired security constraints for the server

3. Click "Start Server"

4. If authentication was chosen the user must now sign in on the server

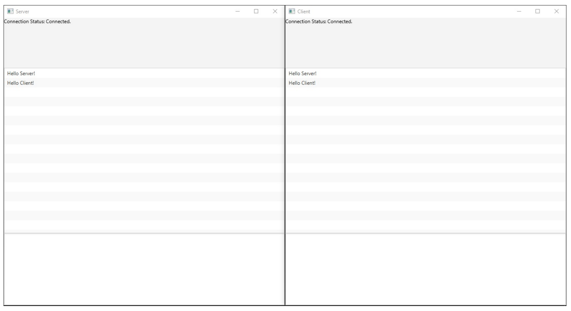

Server
Connection Status: Not connected.
Enter your username and password:
FakeUser
••••••••

4. Select matching security constraints for the client

5. Click "Connect"

6. If authentication was chosen the user must now sign in on the client

7. Start typing and press Enter to send a message



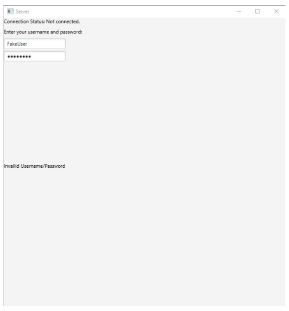
Hello Server!
Hello Client!

Hello Server!
Hello Client!

Test Username and Passwords

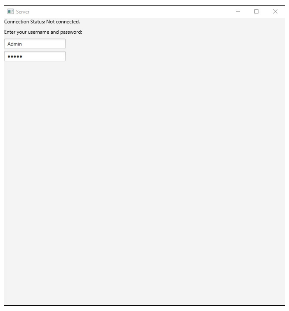
Username: Admin **Password:** admin **Username:** User1 **Password:** password

Screenshots


Server
Connection Status: Not connected.
Enter your username and password:
FakeUser
••••••••

Invalid Username/Password

Screenshot 1: Invalid username


Server

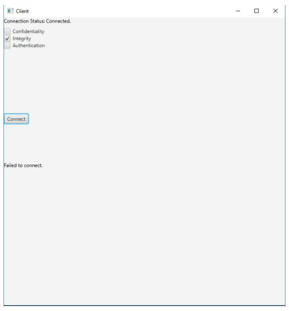
Connection Status: Not connected.

Enter your username and password:

Admin

•••••

Screenshot 2: Server Screen Authentication

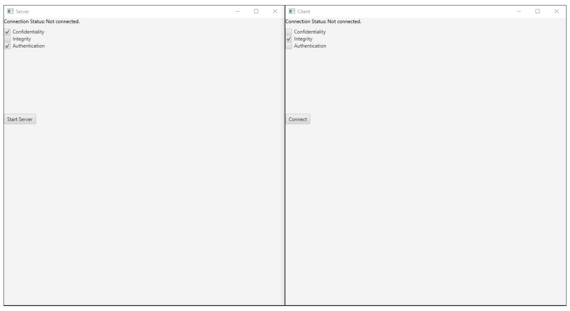

Client

Connection Status: Connected.

- [ ] Confidentiality
- [x] Integrity
- [ ] Authentication

Failed to connect.

Screenshot 3: Client failed to connect


Server

Connection Status: Not connected.

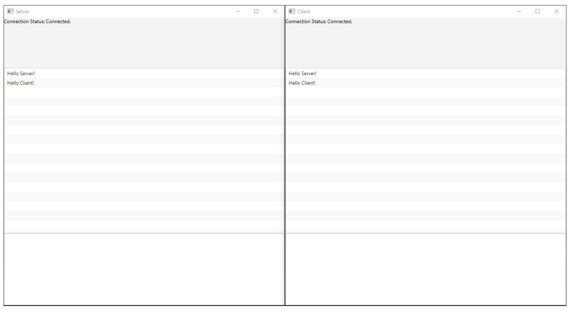
- [x] Confidentiality
- [ ] Integrity
- [x] Authentication

Client

Connection Status: Not connected.

- [ ] Confidentiality
- [x] Integrity
- [ ] Authentication

Screenshot 4: Client and Server don't match


Server

Connection Status: Connected.

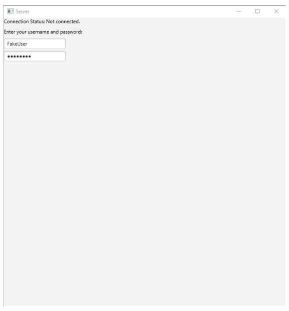
Hello Server!
Hello Client!

Client

Connection Status: Connected.

Hello Server!
Hello Client!

Screenshot 5: Server and Client send messages


Server

Connection Status: Not connected.

Enter your username and password:

FakeUser

••••••••

Screenshot 6: FakeUser login