# SecureIM a encrypted chat client

Arne Bjune and Vegar Engen
{arnedab,vegaen}@cs.ucsb.edu

*Abstract*— **With the recent scandals in government mass surveillance the need for a secure way to communicate becomes clear for anyone that wants to keep their communication private.**

## I. Introduction

In this paper we are going to cover our design for a secure chat client that uses a combination of symmetric AES encryption and RSA public key encryption. Software design choices will be covered inn section II, the authentication protocol in section III and possible improvements in section V.

## II. Software Design

## III. Authentication Protocol

To authenticate the users the server sends a randomly generated challenge the user signs with its private key. This first message is the only message sent in cleartext, the rest is encrypted. Along with the authentication challenge the server supplies its public key, the clients compares it with its stored key, if the clients has no key it accepts and stores the supplied key. The signed challenge from the server is sent back with the clients desired username and public key. The server first checks if it has a public key that corresponds to he username supplied by the client. If a key is found, the server uses the stored key to verify the signed challenge. If no key is found, the supplied key is used and stored for further use.

After both server and client have been authenticated the client sends a request to connect to another user, if the requested user is logged in the server supplies the corresponding public key and establishes a connection. A message sequence chart for the authentication protocol can bee seen in Figure III. All packets are still pass through the server but the content is encrypted with the client's private keys so the server is not able to decrypt it.

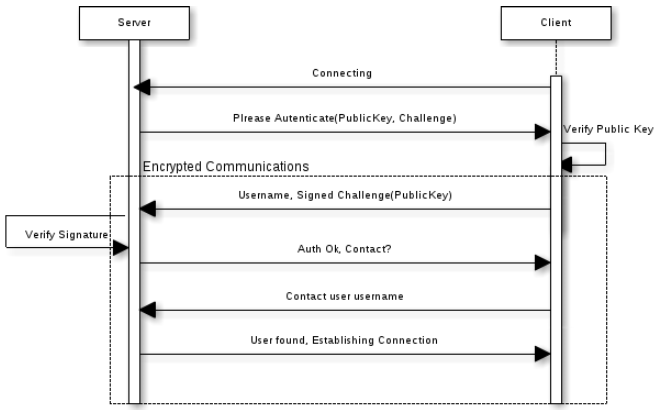

Fig. 1. Authentication process

Even if the clients use public key cryptography the server still has to be trusted to deliver the correct keys and not replacing them with its own. This could be improved by transferring the keys directly from one client to another but this brings up a lot of challenges with devices behind firewalls. However since both the server and client and server are open source you could easily run your own server to provide the needed server trust.

## IV. Conclusion

## V. Improvements

Currently a message going from one client to another needs 4 public key operations and