Discussion

- Motivation
- Overview
- Design
- Advantages
- Disadvantages
- Performance
Motivation

- RSA provides reliable encryption, but..
  - Public Keys are public
  - Keys are time sensitive
- Chat applications generally lack encryption
  - How to detect key compromise?
  - How to handle changing keys?
Overview

- Pseudo Private Public Keys
  - Shared between user and chat server
- Per Session Keys
  - Different Key for each Session
  - Encryption Exponent based on Session
Design (Roles)

Client
- Stores client’s private exponent
- Stores client’s modulus
- Sends server messages encrypted with session public exponent and session modulus

Server
- Supplies client with session public exponent and session modulus
- Decrypts messages with session private exponent and session modulus
- Encrypts messages for clients with each client’s public exponent and modulus
Design (Server Functions)

- Generate List of valid Modulus
  - Ensures all Modulus are relatively prime
- Create client RSA Key
- Assign Session RSA Key on client login
  - Pick random Modulus
  - Generate public exponent based on session
- Relay Messages between Clients
  - Decrypt incoming Message using Session RSA Key
  - Encrypt outgoing Message with each client’s public RSA Key
Design (Client Functions)

- Client Login and Account Creation interface
- Stores Client Private RSA Key and Modulus
- Allows Client to enter messages
  - Encrypts messages with session RSA Key for sending
  - Decrypts messages from server with Client Private RSA Key
- Display messages and list of clients
Advantages

- Encrypted instant messaging
- Increased Security
  - Client RSA Key is pseudo Secret
  - Compromise is limited to current session
- Multiple recipients without increased vulnerability *
Disadvantages

- Increases account creation delay over the life of application
- New Client RSA Key requires new account
- Session exponent limited to session ID
  - 128bit if based on GUID
Performance

- Login and Account Creation
- Messaging
- Encryption Strength