Distributed, Secure Instant Messaging Service (DiSIMS)

By James Luey and Lukas Bischofberger

Problem Statement:

With the rise of Internet censorship and privacy intrusions around the world, security and privacy on the web has become an important topic. Secure messaging systems have emerged as one solution to ensuring privacy between two parties. However most secure messaging systems employ a central server through which all traffic is routed. This poses problems if the server is compromised or inaccessible from the client's location. Our project aims to provide an alternative secure messaging system that has absolutely no central server and therefore avoids these problems associated with current systems.

Project Goal:

To create a secure, distributed, peer to peer messaging system.

Target Customers:

People in countries with heavy censorship

People who want privacy in their online conversations

The average paranoid

Existing Distributed Implementations:

Redact – mobile only

BitTorrent – still in early development

Project Scope:

Only sender and receiver know contents of message

Management of routing table is done without central authority

Outside Project Scope:

Anonymization of sender and receiver

Redirection attacks

Validation of identity

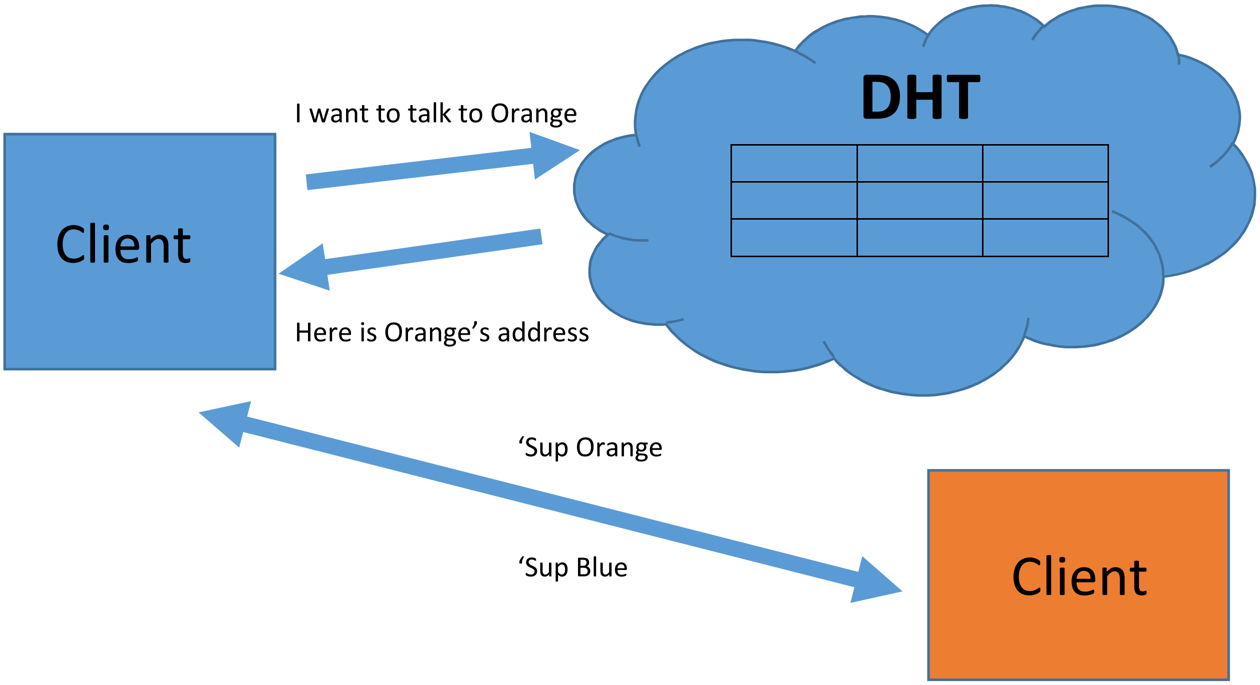
Guarantee of message delivery

Project Outline:

Our implementation would use a distributed hash table to keep track of all the addresses of registered clients. We would use an existing implementation of a distributed hash table (DHT) for our chosen language.

A client would modify the DHT to add an entry for themselves to allow other clients to communicate with it.

To begin a conversation, a client would first look up the address of a desired client in the DHT and try and establish a TCP connection. Upon successful connection, a key exchange is initiated to decide on a symmetric key. From then on all messages are encrypted using that key and decrypted on the other end. We would use an existing encryption scheme such as RSA.



Project Challenges:

- Creating networking code
- Using distributed hash table libraries
- Working with a distributed network
- Security
- Reliable communication