CS 6349 - Network Security – Fall 2016
Programming Project (Draft 3)

Overview

In this project, you will design and implement a secure Internet instant messaging application. Your application will include a client part and a server part. Your program will include a number of security requirements as outlined below. We will evaluate your programs based on their conformance to these requirements.

Supported Functionality

- Clients will have an account with the server to be able to use the application. It is likely that during the account opening phase, the client will obtain necessary security credentials from the server and will use them to do instant messaging with other registered users.
- Clients will have a buddy list and will be able to do instant messaging with these people only. Users can include new registered names into the buddy list or they can remove them from their buddy lists.

Security Requirements

The application will include the following security requirements:

- **Authentication:** An account belongs to an owner who should be the one using it. Unless the account owner shares his/her security credentials with others (which will be a violation of the authentication policy), the use of the security credentials belonging to this account will imply that the user is the owner of the account.
- **Confidentiality:** The messages exchanged between two people will be protected from exposure to others that are not authorized to read what is being communicated.
- **Integrity:** The possible message alteration in transit should not go undetected by the communicating parties.

Project Details

- This is a two-person team project. Please let me know if you have difficulties in finding a partner for your project.
- You will need to complete a project design report and discuss it with the TA. The design report should be submitted by October 28, 2016 but earlier is the better so that you ca n meet the TA and get feedback about your design. The design report will include a summary of the functional capabilities, security features, threat model, and attacks considered when designing the system.
- At the end of the semester, in addition to your program, you will submit a revised project report discussing the above components as they are finally included into the resulting application.

Technical Details

- ~~Mobility of the nodes must be supported, so users cannot be assumed to possess a public key/private key pair.~~
  - We assume that users do not have public/private key pair. Users should be authenticated by using some other mechanism.
- The server possesses a public key/private key pair. Server's public key is known by all users.
- ~~Use a password based authentication protocol for user logins.~~
- ~~You will need to use two keys to provide both integrity and confidentiality of messages; it can be derived from a single shared key using a known pattern.~~
  - The communication between entities should be confidential and integrity protected.
- The actual protocols used by the server and the user is known and specified in advance (e.g. DES in CBC for encryption and SHA or CBC residue for integrity).
- The server is involved only during session establishment between end users; data flow between the users is direct after the session is established.
- Only two users participate in a single session~~, but there may be multiple sessions in parallel for any user with other users~~.
- You do not need to consider the below attack types in your threat model
  - Server break-in attacks
  - Denial of service attacks
  - Identity hiding

Implementation Details

In this project, you will be using various crypto algorithms to implement different components of the application. For this, you will be using the publicly available implementations of well-known crypto algorithm combined into some crypto libraries (check it out at http://www.homeport.org/~adam/crypto/ as an example). However, you are not supposed to use secure transport layer service implementations. That is, you are supposed to build the secure transport layer services by yourselves at the application layer. If you find a library that guarantees authentication+privacy and gives you an API to call it, then you are essentially using an existing transport layer service and this is NOT acceptable.

What to submit

- A design document describing your protocol implementation. It should in general contain how your implementation: 1) performs user logins, 2) allows users to establish a session with another user, 3) provides integrity and confidentiality in message transfer and any other assumptions you have made.
- Soruce code including makefiles and a readme file detailing how to execute your program.