Secure Instant Messaging

Matthew Kramer – Michael Music – Sterling Price
GROUP 5

Application Design

- `client.py`
  - User authentication front-end
  - Send and receive messages
- `server.py`
  - Handles client connections
  - Assigns keys to conversations
- `database.py`
  - Create users in database
  - User authentication back-end
  - Establishes and updates conversation keys
- `diffiehellman.py`

Open Source Modules

bcrypt [3.1.0]
- Password hashing

PyMySQL [0.7.11]
- Communicating with database

PyDES [2.0.1]
- Encryption
- Decryption

Client Authentication

1. User enters their username and password.
2. Database is queried for password hash associated with given username.
3. If username does not exist in the database:
    - Password is salted and hashed.
    - New entry is added to the database.
    - User is authenticated.
4. If username exists in the database:
    - The corresponding hash is retrieved.
    - Entered password is hashed and checked against retrieved result.

Client-to-Client Connection: Server Connection Establishment

The connection between the client and server is a three-step process using a custom application layer protocol:

```
[Client → Server] s:"Alice", d:"Bob", m:"init", sid:0

[Server → Client] s:"Server", d:"Alice", m:"ack_init", sid:1  # User is authenticated

[Server → Client] s:"Server", d:"Alice", m:"ack_init", sid:2  # Destination is online
```

Server can authenticate the client, the client can authenticate the server, and the client knows that the destination user is online.

If the client is not authenticated, the **`m:init`** and **`sid:0`** will not be formatted correctly when received by the server and the connection will be closed.

Client-to-Client Connection: Destination Offline

What if the destination user is not online?

- The client will be added as “logged in” within the server
- The client waits until destination user is logged in, sending periodic messages to probe whether destination user is online

```
[Client → Server] s:”Alice”, d:”Bob”, m:”init”, sid:0
```

- Once user is online, “connection” is created within server:

```
(Alice, Bob, False)
```

Client-to-Client Connection: Key Assignment

Key assignment requires a two-step process:

- Server parses the list of connections and, if any connection contains **`False`**, a new key is generated for use in client communication.
- Generated keys are sent to users in a special ‘keygen’ message format:

```
s:"Server", d:"Alice", m:"KEYGEN-[K_AB]", sid:9999
```

- Both users now have the key to be used for their connection.
- **Note**: at this point, neither user has communicated with the other.

Client-to-Client Connection: Conversation

- Once a key is established, clients now use it to encrypt and decrypt communications.
- The source and destination fields must be unencrypted for the server to route them properly.

[Alice → Server]: s:"Alice", d:"Bob", K_{AB}{m:"Hello Bob!", sid:9}

[Server]: Checks both users are online, destination is not server, message is not init
[Server]: Forward data to socket for Bob

[Server → Bob]: s:"Alice", d:"Bob", K_{AB}{m:"Hello Bob!", sid:9}

- This process is repeated for further messages until either user disconnects from the server.

Encryption and Decryption

Client-to-Client

- Key is negotiated by server and distributed to each user in the conversation
- Once a key is generated, it is updated every 30 seconds with this timer being checked every time a message is encrypted or decrypted using it
- Client-Client encryption uses Triple-DES with 24-byte (192-bit) key

Client-to-Server and Server-to-Client

- Key is exchanged using a Diffie-Hellman key exchange
- Establishes a key shared between the server and client which is used to encrypt all future messages, included those encrypted with client-to-client key

Possible Attacks

1. Will this authenticate Trudy as Alice?

   `[Trudy → Server]: s:"Alice", d:"Server", m:"init", sid:0`

   No, the server will decrypt the message and sid expecting **`m = init`** *and* **`sid = 0`**, *which can not be the case unless Trudy knows Alice's password.*

2. Will this message get forwarded to Bob?

   `[Trudy → Server]: s:"Alice", d:"Bob", m:"Hello Bob!", sid:9`

   No, this data will not be forwarded. Even if it were, Bob will not be able to receive the message, as the formatting will be incorrect due to the end-to-end encryption.

3. Is this system vulnerable to man-in-the-middle attacks?

   No, man in the middle attacks will be impossible, unless encryption is broken or key is obtained.

Fin