Michael Garrett

Pao Fang

Joubin Jabarri

**CSC 250 Programming Assignment**

**Multi-Client/Server Electronic Voting System**

1. **Design Approach and Justification**
   1. Our approach to the design of our voting system put emphasis on CIA. We used AES and RSA encryption/decryption to communicate from each client to the server. We also used keyed hash functions to protect sensitive information such as the voter identification (VID). This provided confidentiality for the voter. We used SHA-256 to provide integrity of the transmitted message to detect bad transmission or tampered ballot. We used a hashed VID
2. **Advantages and Disadvantages**
   1. The advantages to our system are
   2. The disadvantages to our system are
3. **List of Files ( source code and executable )**
   1. README: contains build information
4. **Document Ownership**
   1. Connect.java ( Garrett )
   2. Backend.java (Jabarri )
   3. Frontend.java ( Fang )
5. **Challenges and Lessons Learned**
   1. After we settled on the initial design, our first challenge dealt with how we handled the passing of encrypted/decrypted messages. Initially we were going to use RSA to establish an asynchronous cryptographic connection using the public key of the server. After the client was authenticated we were going to establish a synchronous cryptographic connection using AES and an established session key. However, do the implementation difficulties, we settled on one way communications. The inbound to server connection would be RSA encrypted and the outbound from server connection would be AES encrypted. This way the server always sent a message encrypted with AES and decrypt a message with its private key.
   2. We would definitely add more security to this system. After learning about logging and
   3. In our initial design we also decided to use different languages for different aspects of the system. Although a great learning experience this proved to be rather troublesome. Pao wanted to use java for the front end and Mike wanted to use python for the connection and encryption/decryption. Joubin also wanted to do python for the backend. The problem first arose when we began testing our system. The problem with encrypting in Java and then decrypting in Python is that Java uses PKCS (public key cryptography standard) #1 while Python uses PKCS #8 standard. The padding scheme of both standards are different and there are no solutions that are readily available to make appropriate conversions.