John O'Brien

Joshua Gang

CS419

Project Design Document

3/30/13

Section 1: Protocol

1. User initiates SSL connection to CLA and sends user name.
    1. i. U -> CLA: User Name
2. CLA confirms name is on its list of valid voters.
    1. a. If the voter has already been given a validation number, then terminate the connection.
3. CLA adds user name to list of registered voters.
4. CLA initiates a SSL connection with CTF.
5. CLA generates unique validation number for that voter, and sends it to both the CTF and back to the user.
    - i. CLA -> U: Validation #
    - ii. CLA -> CTF: Validation #
6. The CTF adds this validation number to the list of registered voters who have not yet voted.
7. SSL connection between the user and the CLA is terminated.
8. The user initiates an SSL connection with the CTF.
9. User generates a unique ID, and sends this ID, validation number, and the vote to the CTF.
    - i. U -> CTF: ID | Validation # | Vote
10. The CTF does several checks:
    - a. if the validation number has already been used in the vote, then reject
    - b. if the ID has already been used in the vote, then reject and send a message back to the user to generate a new ID. Go back to step 9
        - i. CTF -> U: Error
11. CTF adds the vote to the appropriate tally, adds the validation number to the list of people who has already voted.
12. After every 5th vote, the CTF sends a list of validation numbers of people who have already voted to the CLA over the same SSL connection. The CLA uses this list and matches them to the given names, allowing people to find out who voted and who didn’t.
    - a. CTF -> CLA: Vote List
13. The election ends when the CTF is told that the election is over. It then sends a list of validation numbers of people who have voted to the CLA, and publishes a list of which

IDs voted for who.

   a. CTF -> CLA: Vote List
   b. CTF -> U: Vote Records

14. The CLA takes the final list of validation numbers and matches them to the given names, saying who voted and who didn’t.
    a. CLA -> U: Vote List with Names

Section 2: Implementation

We write this in python. There is a separate application for the user, CTF, and CLA. The CTF will be told in most cases when the election is over, unless every person has already voted (to be added to protocol later).

Section 3: Proofs

This protects against man-in-the-middle attacks and replay attacks because of the security natures of SSL, so we don’t need to concern ourselves with that.

This protocol only allows authorized users to vote, because of the unique validation number that is generated by the CLA. If someone attempts to vote with an invalid validation number, that was generated by himself, he would have to be extremely lucky to create a validation number that matched one that was already handed out; the vote is only tabulated if the validation number is correct, which the CTF determines from knowledge given to it by the CLA. The CLA only generates the validation number if the requester is an authorized user, so by transitive property only authorized users can vote.

If you attempt to vote more than once with the same validation number, then the CTF is going to reject your vote. You cannot get multiple validation numbers per ID, as the CLA will reject your request for the validation number. Therefore the only way to vote multiple times is to impersonate another authorized user and use their name to request another validation number from the CLA, however that is not possible because of the digital signatures associated with the SSL protocol.

As the list of who has voted and who has not is only published every 5 votes, the best that you can do is determine that a particular ID number is one of five. A malicious user with a valid validation number can generate random IDs and attempt to vote with them; if the CTF says that it is an invalid ID, then he knows that that person has already voted. If the has-voted list is published every one person, then it might be possible to compare lists and what IDs were valid and were not. However, as the list is published every five votes, the best you can do is narrow it down to a 1 in 5 chance of guessing that the ID matches with a certain user. Additionally, if you hit an ID that is not in use, then you cannot use this attack anymore with the same validation number because the CTF will have you registered as having voted already, so it limits the attack by the number of validation numbers present. If one person somehow manages to obtain multiple validation numbers then there is something wrong with the network and the entire election is compromised anyways.

Every voter knows his or her unique ID (unless they forgot it but then shame on them), and when the final list of what IDs voted for what, they can check and make sure that their vote was tabulated.

No one can duplicate anyone else's vote because the IDs of who voted for what are not known until the election has terminated, and the validation numbers are not tied to any vote. Therefor knowing that a validation number has voted does not tell who who it voted for.

This protocol does not protect against both the CTF and the CLA becoming compromised and colluding with one another, but it does prevent either one of them from becoming compromised and screwing over the security of the entire election. For instance, if the CLA becomes compromised, the worst that it can do is say what people's validation numbers are, but then the SSL certificate for that person would also have to be spoofed on the user end of it, otherwise the CLA is not going to accept the vote regardless. It also can't affect the vote of that user because it doesn't handle those. If the CTF becomes compromised, then it can affect the votes themselves, but it doesn't know the users of who voted or not, and at the end of the election the people can go see that something went screwy because they can check themselves. However, if both the CTF and the CLA become compromised, as votes come in, the CTF can just tell the CLA what validation number voted for what, one at a time, and then the CLA knows what those validation numbers match up to , so they can use that to determine the real names of who voted for what.