Alex Vargas, Madrone Goldman

Cryptography Paillier E-Voting

Readme

Hi! Welcome to the readme for our Paillier encryption election board project. Here is what our project does:

* Allows each voter to only vote once (by storing the names of those who have already voted)
* Votes are encrypted with Paillier encryption so that nobody knows who voted for who
* Votes are blind-signed by the EM so we know that the vote going to the bulletin board is the same as that registered with the EM
* Encrypted sums are taken and then decrypted to find the totals for each candidate, and the winner is outputted

Here is what it does not do:

* Zero Knowledge Proof between the voter and the bulletin board. We toiled long and hard over this one but ultimately could not tweak it so that our numbers lined up correctly.

Usage:

1. To compile, you can use “g++ voter.cpp -o [executable name]” and then “./[executable name] to run. Note: this is for Linux based systems. I don’t know the syntax for Windows.
2. A “candidates.txt” file is necessary which needs to contain the number of candidates and the names of all candidates on separate lines. A sample file has been given.
3. The console will prompt a user for their name, and then upon receiving the name of somebody who has not voted will prompt for a number corresponding to a candidate listed onscreen. We do not have a predetermined list of voters, so this name input is the registration with the EM.
4. All voters who wish to vote come up one at a time and input their information and vote. When finished, inputting a “-1” when the program prompts for the voter’s name will stop the input portion of the program. At this point, votes in the bulletin board will be tallied, and the winner will be announced.

Assumptions:

* All voters are registered. They do not need to be on a special voter list.
* Voters cannot see each other’s inputs that may still be visible on the console.

Included Files:

There are 5 files included: voter.cpp, ElectionBoard.h, BulletinBoard.h, primeGenerator.h, and candidates.txt. Voter.cpp contains the main code to run the election and includes the other files in its header. ElectionBoard.h and BulletinBoard.h contain behavior for their respective parts of the election system. PrimeGenerator.h contains a preset list of prime numbers, which is called from the other files to generate suitable prime numbers for encryption. You may notice that these primes are a bit small. Originally, we had very large problems with overflow and we are not implementing a library to use such large numbers, so we decided to scale down a bit.

Other Notes:

* Our blind signature portion uses an RSA blind signature