AnonEvote
A Project on Blockchain E-voting
User Manual
Supervisor: David Gray

Michael Wall
13522003

Monday 22nd May, 2017

Abstract

The AnonEvote system is a proof of concept for a distributed trustless voting system, that provides user anonymity and tamper-proof ballots. It makes use of Homomorphic cryptography and a blockchain database on a peer to peer network to achieve this functionality.

Contents

1 Installation

1.1 Installing Go and Git

It is assumed that the user has at least version 1.8 installed, and has set up their Go workspace appropriately. More details on installing Go and setting up environment variables can be found here: `https://golang.org/doc/install`.

The installation of Go can be verified by running:

```
thewalla07:~/work...tation/res: go version
go version go1.8 linux/amd64
```

It is also assumed that the user has Git installed on their machine. A user can verify this by running the following command:

```
thewalla07:~/work...tation/res: git version
git version 2.13.0
```

1.2 Installing the voting system

The system should be cloned using Git as follows:

```
thewalla07:~/example: git clone https://github.com/CPSSD/voting
...
thewalla07:~/example: cd voting/src/
thewalla07:~/exam...voting/src:
```

2 Election generation

For this proof of concept project, an election can be set up as follows. Once in the src directory for the project, run the generator and follow the instructions on screen. An example run through is shown here:

```
thewalla07:~/exam...voting/src: go run generator/generate.go
Number of voters to generate: 12
Threshold number of shares to construct election key: 9
Allow peers to sync? (y/n): y
```

```
Minimum known starting peers? (min recommended = 1): 5
Initial port number for nodes: 8100
Number of characters in a vote node: 20
How many selections are on the ballot? 3
Use double quotes for description entries
Enter user description for selection 1: "Alice"
You entered: Alice
Enter user description for selection 2: "Bob"
You entered: Bob
Enter user description for selection 3: "Eve"
You entered: Eve
Building election config...
Done
thewalla07:~/exam...voting/src: ls *.json
0.peer.json 1.peer.json 2.peer.json 4.peer.json 6.peer.json 8.peer.json
10.peer.json 11.peer.json 3.peer.json 5.peer.json 7.peer.json 9.peer.json
thewalla07:~/exam...voting/src:
```

Generation of the user's DSA keys may take a while. Once the generation is complete, you will be able to see the specified number of user node configuration files in the src directory.

3 Running a node

A node for a user can be run as follows:

```
thewalla07:~/exam...voting/src: go run main.go 0.peer.json
Welcome to voting system.
Your vote token is: BpLnfgDsc2WD8F2qNfHK
What next? (h for help):
```

3.1 Viewing help

A list of commands can be found by typing "h":

```
What next? (h for help): h
    h       Print this help
    peers       Print known peers
```

```
    pool        Print pool of transactions
    chain       Print current chain
    v       Cast a vote
    q       Quit program
    b       Broadcast share
    r       Reconstruct election key
    tally       Tally the votes
What next? (h for help):
```

3.2 Viewing peers

Typing peers will yield the following output:

```
What next? (h for help): peers
Peers:
localhost:8110
localhost:8109
localhost:8106
localhost:8111
localhost:8100
localhost:8101
localhost:8102
localhost:8104
localhost:8107
localhost:8103
localhost:8105
localhost:8108
```

3.3 Casting a vote

```
What next? (h for help): v
Enter your selection (0 or 1) for Candidate Alice: 0
Enter your selection (0 or 1) for Candidate Bob: 1
Enter your selection (0 or 1) for Candidate Eve: 0
```

3.4 Current pool of transactions

```
What next? (h for help): pool
```

```
 // Vote Token:    BpLnfgDsc2WD8F2qNfHK

 // Vote Token:    5a84jjJkwzDkh9h2fhfU

 // Vote Token:    VuS9jZ8uVbhV3vC5AWX3
```

3.5 Handling bad inputs

```
What next? (h for help): badinput
Unrecognised input
```

3.6 Viewing the current chain

```
What next? (h for help): chain
Entering print chain
Chain:
Block 0:
 // Proof of Work: 00000654657dce58341f0e474f7954...
 // Parent Proof:  00000000000000000000000000000000

Transaction 0:
 // Vote Token:    BpLnfgDsc2WD8F2qNfHK
Transaction 1:
 // Vote Token:    5a84jjJkwzDkh9h2fhfU
Transaction 2:
 // Vote Token:    VuS9jZ8uVbhV3vC5AWX3
Transaction 3:
 // Vote Token:    9IVUWSP2NcHciWvqZTa2

Block 1:
 // Proof of Work: 00000a40b0a3e2f2d35e7923165fcc...
 // Parent Proof:  00000654657dce58341f0e474f7954

Transaction 0:
 // Vote Token:    N95RxRTZHWUsaD6HEdz0
Transaction 1:
 // Vote Token:    ThbXfQ6pYSQ3n267l1VQ
Transaction 2:
```

```
 // Vote Token:    KGNbSuJE9fQbzONJAAwd
Transaction 3:
 // Vote Token:    CxmM8BIabKERsUhPNmMm

Block 2:
 // Proof of Work: 00000e559ea7eb59049af102d7492e...
 // Parent Proof:  00000a40b0a3e2f2d35e7923165fcc

Transaction 0:
 // Vote Token:    df2eSJyYtqwcFiUILzXv
Transaction 1:
 // Vote Token:    2fcNIrWO7sToFgoilA0U
Transaction 2:
 // Vote Token:    1WxNeW1gdgUVDsEWJ77a
Transaction 3:
 // Vote Token:    X7tLFJ84qYU6UrN8ctec


Exited print chain
```

3.7 Reconstructing the election key

A user may attempt to reconstruct the election key at any stage as shown below.

3.7.1 No shares available

```
What next? (h for help): r
Attempting to reconstruct the election key
0
0
What next? (h for help): tally
Calculating the tally...
Totals for the election are as follows:
Alice: 0 votes
Bob: 0 votes
Eve: 0 votes
```

3.7.2 Some shares available

A user will be able to see the current representation of the two secret exponents of the election private key. On attempting to decrypt the tally without enough votes will yield a nonsense result. Note the output is cut short for the purposes of this manual.

```
What next? (h for help): b
Broadcasting our share of the election key
What next? (h for help): r
Attempting to reconstruct the election key
29594219322160163621004745019466797508041948621...
60066052256834636224488809670249179403617893133...
What next? (h for help): tally
Calculating the tally...
Totals for the election are as follows:
Bob: 6340704023108678213931106317521312645936O1.... votes
Eve: 11927441511688941042897411593329679710059 6... votes
Alice: 13407502637355031058454296419343066565 06... votes
```

3.7.3 Successful reconstruction and tally

Once enough shares have been distributed, we can reconstruct the secret values of the private key. When we run the tally, we can see the results of the election.

```
What next? (h for help): r
Attempting to reconstruct the election key
13645343740650522540127563557632198543982355422...
43572813250040380242669599819834001774640728225...
What next? (h for help): tally
Calculating the tally...
Totals for the election are as follows:
Alice: 4 votes
Bob: 4 votes
Eve: 4 votes
```

3.8 Quiting the program

```
What next? (h for help): q
```

```
Waiting for processes to quit
thewalla07:~/exam...voting/src:
```