Verification Server

Operating System: Ubuntu

AWS or local installation

The Verification, QR Code, and Caching server can either be all installed on separate computers, or all run on the same computer. These instructions will outline how to set up the Verification server as a separate server, and at the end there will be instructions on how to modify the server if you want to run them all on the same computer for simplicity.

Each Node.js server requires `npm` be installed, as well as all the dependencies for the project.

If you have already set up the QRCodeServer and intend on running all Node.js servers on the one computer, then skip to the end of this document.

Install npm

The first step is to install npm so that the Node.js servers can run. Run the following command in a new terminal

```
sudo apt-get install npm
```

We now need to download the web applications from GitHub. Perform the following command in a terminal in your `home` directory.

```
git clone https://github.com/SwinburneBlockchain/WebApplications
```

This copies the Node.js web server from the GitHub link. If you plan on running each Node.js server on individual servers then you may delete the `CachingServer.js` and `QRCodeServer.js` files that were downloaded along with the `VerificationServer.js` file. Otherwise continue onto the next step.

To initialise this application, you will need to move into the WebApplications folder created when you cloned the GitHub link, and initialise npm in this folder.

```
cd WebApplications
npm init
```

You will now need to edit the `package.json` file that was downloaded from GitHub. Open up the `package.json` file, and delete the following line:

```
"mongodb": "^3.2.16"
```

Now follow these instructions on installing MongoDB on Ubuntu

https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-16-04

If you are having trouble starting MongoDB after following these instructions, see Appendix 1.

You should now be able to run the following command which installs the remaining dependencies for the web server:

```
npm install
```

Once everything has been installed, run the Verification Server by entering the following command:

```
node VerificationServer.js
```

Modifying Program Constants

The Verification Server holds the publicly available information about the Proof-of-Location Bluetooth beacons. You will need to change the RSA PEM public keys and locations of the producer Bluetooth beacons. You will need to enter the ones you generated into this array, in the following format:

```
['RSA public key', 'location']
```

You should have one for each producer in the system.

```
21  var locationServers = [
22    ['-----BEGIN PUBLIC KEY-----MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8
23    ['-----BEGIN PUBLIC KEY-----MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8
24    ['-----BEGIN PUBLIC KEY-----MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8
25  ];
```

Running all Node.js servers on the same computer

There are several modifications that need to be made to the `VerificationServer.js` file if you wish to run all the servers on the same computer.

The first change is that you will need to comment out lines 63 - 68. This is the part which starts the Node.js server on port 3000. You cannot have multiple Node.js servers running on port 3000.

```
62
63    var port = process.env.PORT || 3000;
64      app.listen(port, function () {
65        console.log('VerificationServer - Listening on port 3000...')
66      });
67
68  });
69
```

Next you will need to change line 78 so that it reads `router.post` instead of `app.post`.

```
74  /*
75    Verifies that the location proof associated with a MOVE action is valid.
76    Takes in generated hash, the RSA public key, the 'location proof' (signature)
77  */
78  app.post('/verifyLocation', function(req, res) {
79      var fullHash = req.body.hash;
80      var publicKey = req.body.publicKey;
81      var locationProof = req.body.locationProof;
82      var timestamp = req.body.timestamp;
```

Updated.

```
74  /*
75    Verifies that the location proof associated with a MOVE action is valid.
76    Takes in generated hash, the RSA public key, the 'location proof' (signature)
77  */
78  router.post( /verifyLocation', function(req, res) {
79      var fullHash = req.body.hash;
80      var publicKey = req.body.publicKey;
81      var locationProof = req.body.locationProof;
82      var timestamp = req.body.timestamp;
```

That is all the modifications required to the `VerificationServer.js` file. Once all additional Node.js servers have been set up, the program can be run by running `node QRCodeServer.js` from the terminal.

Appendix 1 – Trouble with MongoDB

This may resolve issues with starting MongoDB

Open a terminal and enter the following command:

```
sudo nano /etc/systemd/system/mongodb.service
```

Now copy the following text into the newly created file:

```
[Unit]
Description=High-performance, schema-free document-oriented
database
After=network.target

[Service]
User=mongodb
ExecStart=/usr/bin/mongod --quiet --config /etc/mongod.conf

[Install]
WantedBy=multi-user.target
```

You should now be able to start MongoDB through the following command:

```
sudo systemctl start mongodb
```

To permanently enable Mongodb, enter the following command:

```
sudo systemctl enable mongodb
```