AnonPeerster

Dependencies

This project depends on **DeDiS Protobuf** and **go-sqlite3**, which can be installed by running:

```
go get github.com/dedis/protobuf
go get github.com/mattn/go-sqlite3
go install github.com/mattn/go-sqlite3
```

Note that installing **go-sqlite3** requires gcc (both on Linux and on Windows), since it is a cgo package.

How to run

After compiling the package with `go build` and renaming the executable "Project" to "gossiper", you can run `gossiper -h` to print the list of command-line arguments.

Mandatory arguments

- `-dataDir=...` the directory for storing the SQLite3 database and RSA keypair. If the directory does not exist, it will be created (along with an empty database and a new keypair/identity).
- `-gossipAddr=...` address/port for the gossiper socket. You can specify a full IP address:port like `127.0.0.1:5000` to listen on a specific interface, or `:5000` to listen on all interfaces.

Optional arguments

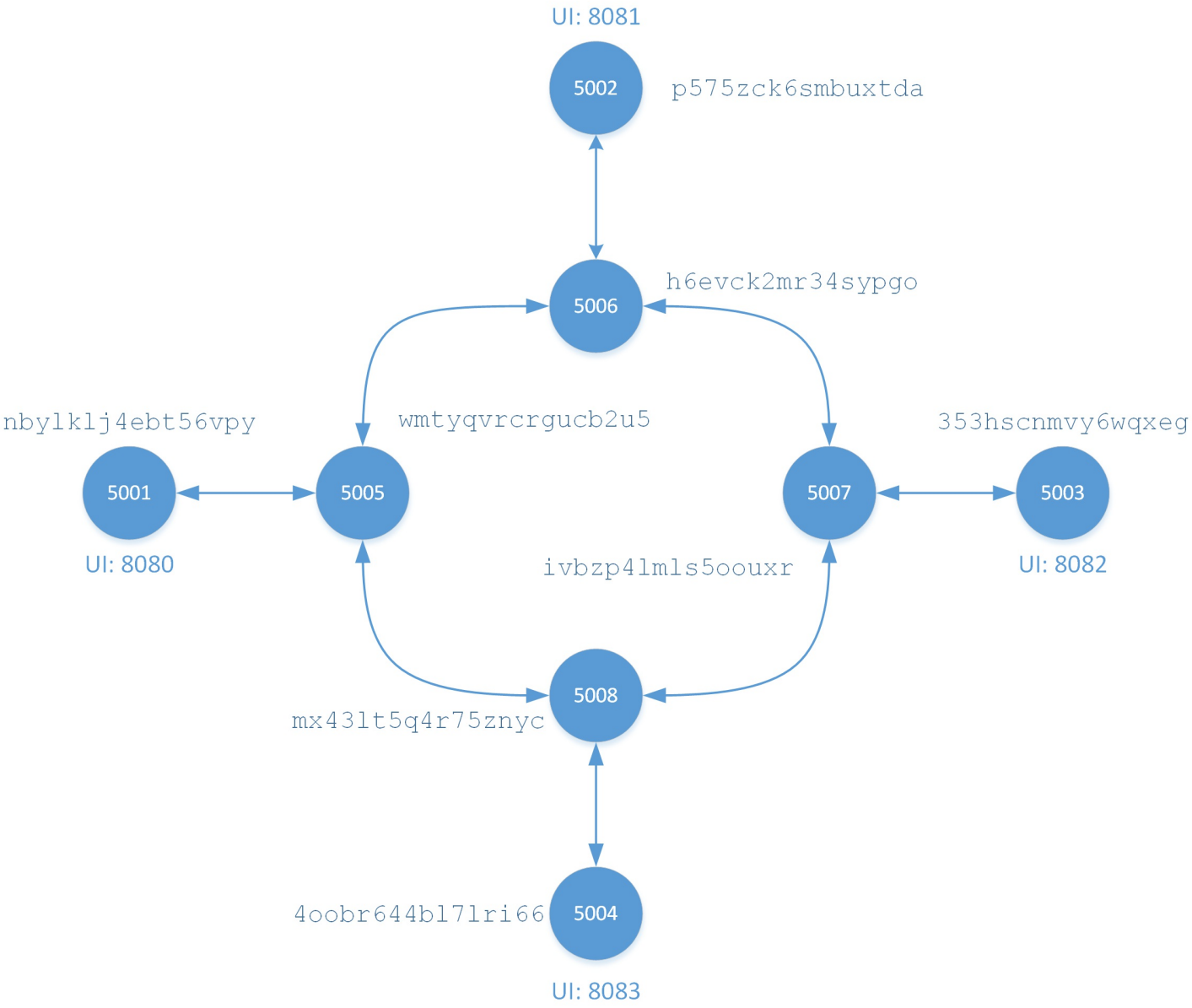
- `-peers=...` peers separated by commas.
- `-UIPort=...` port for the HTTP client, which listens only on `localhost`.
- `-powDifficulty=...` proof-of-work difficulty (default: 18 leading zeros).

Example

```
gossiper -dataDir=_data/RingA -gossipAddr=:5005 -peers=127.0.
0.1:5006,127.0.0.1:5008,127.0.0.1:5001 -UIPort=8080
```

Test scripts

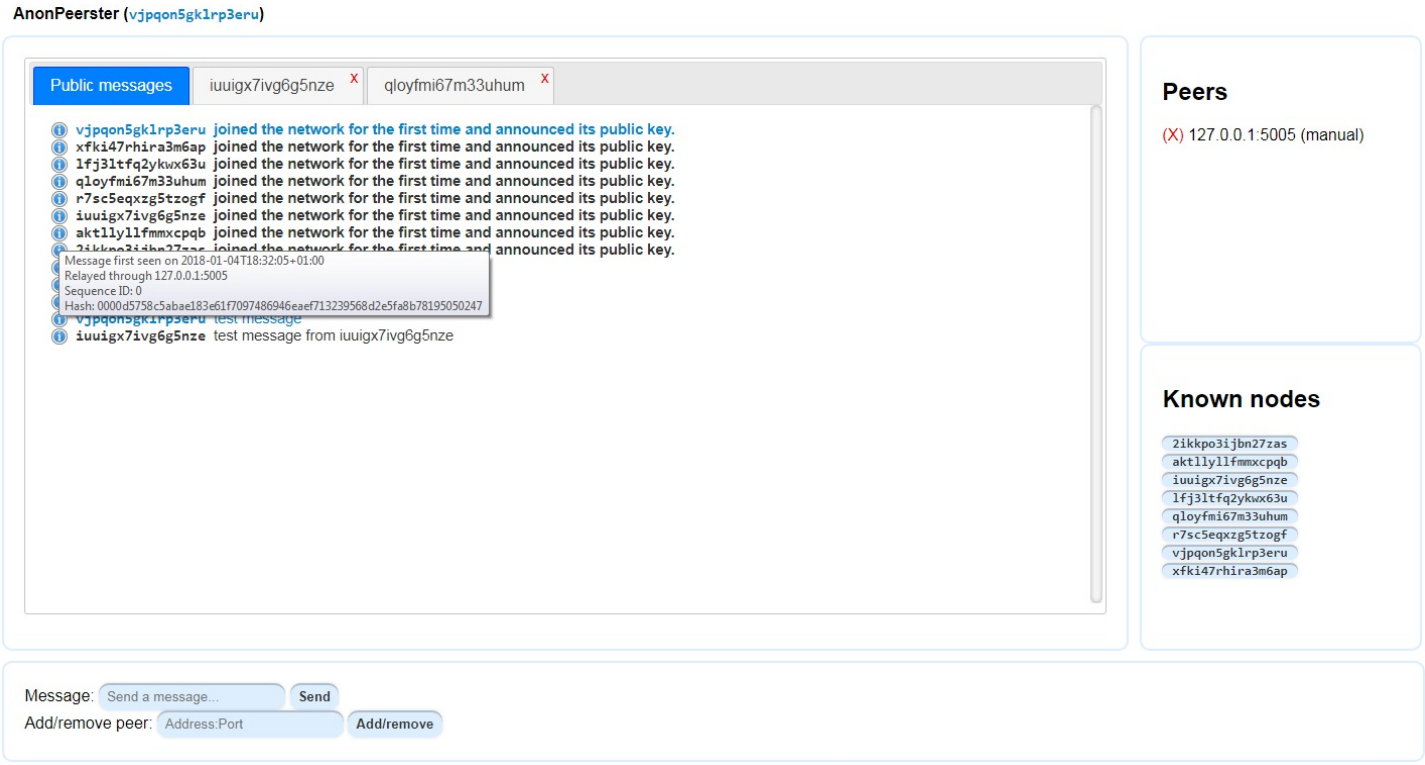
We have included a test script `ring_test.sh` and `ring_test.bat` (respectively for Linux and Windows). It creates ring network topology with 8 nodes, as shown in the figure below:



The keys and databases for these nodes are stored in the `_data` directory. Of course, you are free to delete these files for your tests. If you delete the `key.bin` file (which contains the RSA keypair), the application will generate a new identity. If you delete the SQLite3 database `messages.db`, it will create a new empty database and synchronize messages as usual. The SQLite database can be opened by any standard SQLite database explorer.

User interface

You can access the user interface through `http://localhost:UIPort`.
Note that the browser must support the latest JavaScript standard (ES6).



With the GUI, you can:

- Send a message to the public room (unencrypted, but signed).
- Open a private chat with one of the known nodes and send a private message (encrypted and signed).
- Show additional information about a message (e.g. its hash) by hovering over the **(i)** icon.
- Add/remove peers.

The console shows some informative messages, such as the proof-of-work status when sending a new message:

```
PUBLIC MESSAGE FROM CLIENT: test
```

```
INFO: starting a nonce computation with 16 leading zeros...
INFO: nonce computed in 0.54 seconds (218522 tries)
```