How to volunteer by helping I2P-Bote bootstrap

An easy way to help people message eachother privately is to run an I2P-Bote peer which can be used by new bote to bootstrap their own I2P-Bote peers. Unfortunately, until now, the process of setting up an I2P-Bote bootstrap peer has been much more obscure than it should be. It's actually extremely simple!

```
What is I2P-bote?
   I2P-bote is a private messaging system built on i2p, which has additional
   features to make it even more difficult to discern information about the
   messages that are transmitted. Because of this, it can be used to transmit
   private messages securely while tolerating high latency and not relying on
   a centralized relay to send messages when the sender goes offline. This is
   in contrast to almost every other popular private messaging system, which
   either require both parties to be online or rely on a semi-trusted service
   which transmits messages on behalf of senders who go offline.

   or, ELI5: It's used similarly to e-mail, but it suffers from none of e-mail's
   privacy defects.
```

Step One: Install I2P-Bote

I2P-Bote is an i2p plugin, and installing it is very easy. The original instructions are available at the bote eepSite, bote.i2p, but if you want to read them on the clearnet, these instructions come courtesy of bote.i2p:

1. Go to the plugin install form in your routerconsole: http://127.0.0.1:7657/configclients#plugin
2. Paste in the URL http://bote.i2p/i2pbote.su3
3. Click Install Plugin.
4. Once installed, click SecureMail in the routerconsole sidebar or homepage, or go to http://127.0.0.1:7657/i2pbote/

Step Two: Get your I2P-Bote node's base64 address

This is the part where a person might get stuck, but fear not. While a little hard to find instructions, this is actually easy and there are several tools and options available to you, depending on what your circumstances are. For people who want to help run bootstrap nodes as volunteers, the best way is to retrieve the required information from the private key file used by the bote tunnel.

I2P-Bote stores it's destination keys in a text file which, on Debian, is located at /var/lib/i2p/i2p-config/i2pbote/local_dest.key. In non-Debian systems where i2p is installed by the user, the key will be in $HOME/.i2p/i2pbote/local_dest.key, and on Windows, the file will be in C:\ProgramData\i2p\i2pbote\local_dest.key.

Method A: Convert the plain-text key to the base64 destination

In order to convert a plain-text key into a base64 destination, one needs to take the key and separate only the destination part from it. In order to do this properly, one must take the following steps:

1. First, take the full destination and decode it from i2p's base64 character set into binary.
2. Second, take bytes 386 and 387 and convert them to a single Big-Endian integer.
3. Add the number you computed from the two bytes in step two to 387.
4. Take that nummber of bytes from the front of the full destination.
5. Convert back to a base64 representation using i2p's base64 character set.

A number of applications and scripts exist to perform these steps for you. Here are some of them, but this is far from exhaustive:

- the i2p.scripts collection of scripts(Mostly java and bash)
- my application for converting keys(Go)

These capabilities are also available in a number of I2P application development libraries.

Shortcut:

Since the local destination of your bote node is a DSA destination, then it's quicker to just truncate the local_dest.key file to the first 516 bytes. To do that easily, run this command when running I2P-Bote with I2P on Debian:

```
sudo -u i2psvc head -c 516 /var/lib/i2p/i2p-config/i2pbote/local_dest.key
```

Or, if I2P is installed as your user:

```
head -c 516 ~/.i2p/i2pbote/local_dest.key
```

Methon B: Do a lookup

If that seems like a bit too much work, it's possible for you to look up the base64 destination of your Bote connection by querying it's base32 address using any of the available means for looking up a base32 address.

Step Three: Contact Us!

Update the built-in-peers.txt file with your new node

Now that you've got the correct destination for your I2P-Bote node, the final step is to add yourself to the default peers list for I2P-Bote here here. You can do this by forking the repository, adding yourself to the list with your name commented out, and your 516-char destination directly below it, like this:

```
    # idk
    QuabT3H5ljZyd-PXCQjvDzdfCec-2yv8E9i6N71I5WHAtSEZgazQMReYNhPWakqOEj8BbpRvnarpHqbQjoT6yJ5UObKv2hA2M4XrroJmydPV9CLJUCqgCqFfpG-bkSo0gEhB-
GRCUaugcAgHxddmxmAsJVRj3UeABLPHLYiakVz3CG2iBMHLJpnC6H3g8TJivtqabPYOxmZGCI-P~R-
s4vwN2st1lJyKDl~u7OG6M6Y~gNbIzIYeQyNggvnANL3t6cUqS4v0Vb~t~CCtXgfhuK5SK65Rtkt2Aid3s7mrR2hDxK3SIxmAsHpnQ6MA~z0Nus-
VVcNYcbHUBNpOcTeKlncXsuFj8vZL3ssnepmr2DCB25091t9B6r5~681xGEeqeIwuMHDeyoXIP0mhEcy3aEB1jcchLBRLMs6NtFKPlioxz0~Vs13VaNNP~78bTjFje5ya20ahWlO0Md~x5P5lWl
```

and submitting a pull request. That's all there is to it so help keep i2p alive, decentralized, and reliable.