An Introduction to Anonymous Communication with I2P

To be or I2P

Jens Kubieziel <jens@kubieziel.de>

2007-12-27

Abstract Many of you may know about Tor or JonDo. These are widely deployed anonymising systems. Another promising approach is I2P. This paper will show the basic concepts of this network and introduce some applications.

1 Introduction

Anonymous communications are getting more important nowadays. On the one hand are companies which try to invade your privacy by using several well-known techniques (i. e. Cookies, JavaScript). These are used to build individual profiles of your behaviour and to send you better crafted spam. ☺ The government, on the other hand, creates laws (e. g. the data rentention law) designed to help improve law enforcement. But they can easily be abused to spy on you. And several "interested third parties" have declared a strong interest in the data gathered in this way. Therefore users see an increased need for protection against traffic analysis.

At past Chaos Communication Congresses, several solutions have been presented. There were remailers like Mixmaster[1] or Mixminion[2] as well as the anonymous network Tor[3] introduced. One interesting approach has however not yet been mentioned. The I2P[4] anonymous network tries to build VPN-like connections between its participants using a P2P-approach. The following document will give you a short overview of I2P. If you want a more detailed view of I2P's working principles have a look at the documents at the above mentioned website.

2 Nomenclature

I2P uses a special nomenclature for some parts of their protocol. To better understand the following it is important to know about it.

router Software which participates in the network.

tunnel A path through several routers which is used to transport encrypted packets.

inbound and outbound tunnel Every tunnel in I2P is unidirectional. The tunnel

[1] http://mixmaster.sourceforge.net/
[2] http://mixminion.net/
[3] https://www.torproject.org/
[4] http://www.i2p.net/

for incoming connections is called the inbound tunnel and the one for outgoing connections is called the outbound tunnel. A router usually has several inbound and outbound tunnels.

tunnel gateway This collects messages, does some preprocessing, encrypts the data and sends it to the next router. A gateway of an outbound tunnel is the creator of that tunnel. The gateway of an inbound tunnel receives messages from any peer and forwards them until they reach the creator.

endpoint The endpoint of a tunnel is either the creator (inbound) or the last hop of that tunnel (outbound). In the case of an outbound tunnel the endpoint is not necessarily the desired location. In fact, the endpoint looks for another tunnel gateway to send the packets along.

netDb is the short name for network database. It is a pair of algorithms which are used to share the network metadata. It gives your router all necessary data to contact other routers.

As you can see there is no client, server or exit nodes—in I2P every router can be client and server. It forwards packets from your computer as well as for other computers. Furthermore *all* communication stays within the I2P-network[5] and is end-to-end encrypted. A router doesn't know about its role and as the message is encrypted it has no possibility of learning about its contents.

[5]There are proxies for non-I2P communication.

3 Anonymous communication with I2P

What happens exactly if Alice wants to send a message to Bob? First, Alice's router must know how to reach Bob's. She asks the netDb for Bob's `leaseSet`. This is special metadata and gives Alice's router the gateways of Bob's inbound tunnels plus other information. Now Alice picks one of her outbound tunnels and sends it. The message has instructions for Alice' endpoint on how to forward the message to Bob's inbound gateways. The endpoint forwards the message as requested and Bob's gateway forwards it to Bob's router. If a reply from Bob to Alice's message is desired, Alice's destination is also sent in her message, so saving Bob from performing a netDb lookup.

This is the basic working principle of I2P. The following sections will show you details of I2P's components.

3.1 netDb

The network database, called netDb, shares network metadata consisting of a pair of algorithms. First there is a small set of routers called "floodfill peers". The rest of the routers participate in a special algorithm, Kademlia.

3.1.1 Network metadata

There are two types of network metadata: `routerInfo` and `leaseSet`.

The `routerInfo` structure supplies routers with the data necessary for contacting a particular router. It contains their public keys (2048 bit ElGamal, 1024 bit DSA plus a certificate), the transport address (IP address and port) and some arbitrary uninterpreted text options. All of this

information is signed with the included DSA key.

The other structure `leaseSet` is similar in some ways. It also contains the public keys (ElGamal, DSA and certificate) and includes a list of leases and a pair of public keys for encrypting messages to the destination. The leases specify one of the destination inbound tunnel gateways. This is achieved by including the SHA-256 hash of the gateway's identity, a 4 byte tunnel id and the expiration time of that tunnel.

3.1.2 Bootstrapping

How is the netDb initially built? A router needs at least one `routerInfo` of a reachable peer. It then queries that peer for references for other routers and uses the Kademlia healing algorithm. Each `routerInfo` reference is stored in an individual file in the router's netDb subdirectory. This allows these references to be easily shared, so bootstrapping new users.

3.2 Tunnels

As described above tunnels are unidirectional and consist of an inbound and an outbound tunnel. Both work along similar principles. They have a gateway, an endpoint and (probably) some routers inbetween. The gateway collects messages and performs some preprocessing. After these initial steps it encrypts the data and sends it to the first router in the tunnel. All subsequent routers check the integrity of the message and add a layer of encryption. At some point the message arrives at the endpoint, where it is forwarded as requested.

4 Applications

As you have seen I2P is an anonymous IP layer. What applications could you use with I2P? The developers have implemented several commonly-used programs. At the moment, programs for mail, websites, chat, filesharing and more exist. For most of these tasks, special programs are needed as commonly available software has no support for I2P.

4.1 Websites

Websites in I2P are called *eepsites* and have the top level domain `.i2p`. To visit an eepsite, point your browser's proxy to port 4444. Your local I2P client handles the request. Unlike Tor's hidden services, all eepsites use readable names. You can reach the eepsite of I2P via `http://www.i2p/` and I2P's forum at `http://forum.i2p/`.

If you want to provide information at your own eepsite, you must follow several steps:

1. pick a lowercase name for your eepsite
2. start the eepsite at your I2P configuration window and configure it
3. add content to `i2p/eepsite/docroot`
4. add your site to an I2P address book (`http://orion.i2p/` or `http://trevorreznik.i2p/`)
5. wait for your first visitor ☺, additionally you can make your site public by posting to the forum, to the wiki or telling others about it in IRC

Additionally you can browse to websites outside of I2P. Just set your local

HTTP proxy to localhost with port 4444 and enter "normal" domain names.

4.2 Email

For email there is a web interface or you can also use your mail client. An email address in i2p has the form `username@mail.i2p`. The username can be freely chosen. Just go to the Postman HQ[6] and create a new mailbox. This site also has instructions on how to setup your mail client. Once you are ready, you can send emails. Another way to send your emails is to use the web interface called *Susimail*. Just log on with your username and password.

You can also use I2P to communicate with the outside world. I2P mail can connect to an internet mail server[7] where it rewrites your email address with `username@i2pmail.org`. The receiver can answer it. The mail server will restore the domain name to `mail.i2p` and forward it to your mailbox.

4.3 Blogging

Syndie is a censor resistant, anonymous blogging tool. You can write postings which are then published on your local pc and on distributed archives. The software is not part of the I2P distribution. It can be downloaded from `http://syndie.i2p/` and, like I2P, is written in Java. After installation is finished, the software has to be configured. If you only want to read other postings, you can subscribe to the forum. In case you also want to publish blog postings, more work must be done. First choose a nickname, then choose how Syndie connects to archive servers and in the end add any desired forums. Syndie contains a button labelled `Post`. Click on it and write your postings.

4.4 Chat

The main chat protocoll is IRC. Point your chat client to localhost with port 6668 and choose a channel.

4.5 File sharing

There are several clients for several networks. I2PSnark is bundled with I2P and offers you access to Bittorrent. Furthermore the developers of Azureus have written azneti2p, which is also a Bittorrent client. I2Phex is a port of the Phex Gnutella client and, lastly, IMule allows access to eMule.

[6] `http://hq.postman.i2p/`
[7] `mx.i2pmail.org`