I2P - The Invisible Internet Project

Felipe Astolfi
Leiden University
fastolfi@gmail.com

Jelger Kroese
Leiden University
jelgerkroese@gmail.com

Jeroen van Oorschot
Leiden University
post@jeroenvanoorschot.nl

ABSTRACT
I2P is an open source Internet technology that makes it possible to use and provide Internet services anonymously. It does so by creating a hidden network within the Internet, a Darknet. This report looks into the history and development of the software, the way I2P works and provides a short developers guide for setting up a website within the I2P network. Next to that, some applications of I2P are covered, including anonymous emailing, web hosting and file sharing. Furthermore, I2P is compared to another anonymizing internet technology called Tor by looking at their strengths and weaknesses with regard to different applications. It is concluded that I2P currently offers high levels of anonymity and is quicker than Tor for some applications within a Darknet. Though, for accessing the regular Internet anonymously, Tor proves to be a better option. However, I2P is still beta software which requires more development. With more research, funding and a larger user base it could potentially provide a faster and more secure way of using Internet services anonymously than is currently offered by other applications.

1. PURPOSE, CONTEXT AND HISTORY

The Invisible Internet Project (I2P) is an anonymous network that can be accessed through regular web browsers. It acts as a layer upon the internet, which provides users the possibility to anonymously exchange data. In this way it creates a network within the Internet. All aspects of the network are open source, so it can be used to both create web services and to use web services anonymously. The network has no central point of communication, which means that there is no central point where security or anonymity can be compromised. Since absolute anonymity does not exist, there is always a tradeoff between security protection and the computational power needed to make or break the security. In I2P, users are in control of the level of security, anonymity, bandwidth usage and latency to meet their specific needs [5].

I2P has been a beta release since its development started in 2003. It has been constantly updated between then and now, but its founders still claim that it needs more peer reviews and research for a stable release.

- February 2003: I2P proposed as modification to Freenet, another anonymous network [10].
- April 2003: I2P grows into own platform as 'anonCommFramework' [10].
- July 2003: anonCommFramework becomes I2P [10].
- August 2003: start of writing code [10].
- March 2014: Internet search company DuckDuckGo awards $5000 to I2P, as part of their open-source donation program [6].
- August 2014: launch of the Privacy Solutions project, a new organization that develops and maintains I2P software [6].
- August 2014: Privacy Solutions releases I2P Android on Google Play [6].

2. OPERATING PRINCIPLES

The I2P network consists of a group of virtual routers. A router is a piece of software that enable applications to communicate with each other through the network [14]. The key point that ensures anonymous communication is the fact that sender and receiver do not communicate with each other directly, but through multiple other routers. Everybody using the network acts as a router as well, so each router constantly sends and receives a multitude of messages. Some of these messages might be directed to or coming from that specific user, but in most cases it will just act as a hop to send other people's communication through. Since these types of communication are undistinguishable it is nearly impossible to see what communication takes place between which persons in the network. Thus anonymity is achieved.

2.1 ROUTING MESSAGES THROUGH I2P

I2P can be seen as a secure and anonymous IP layer, where instead of addressing messages to IP addresses, they are addressed to *location independent identifiers*. Also the messages can be significantly larger than IP packets [6].

All communication between routers goes through tunnels. A tunnel is a unidirectional path through multiple routers. The outbound tunnel is a path that is only used to send messages away from the tunnel creator, the inbound tunnel is a path that is only used to send messages towards the tunnel creator [2]. Every router has multiple inbound and outbound tunnels. Figure 1 shows how a message is transferred from one person to another through the I2P network. In order to send a message from Bob to Alice, it first goes through Bob's outbound tunnel. At the end of Bob's outbound tunnel it enters Alice's inbound tunnel (possibly after passing through more routers), which eventually delivers the message to Alice. By using separate tunnels for incoming and outgoing messages, both Bob and Alice can set the minimum amount of hops (routers) they want to use for their communication. This ensures that communication will always meet the

minimum security level they require, while still operating under workable latency levels.

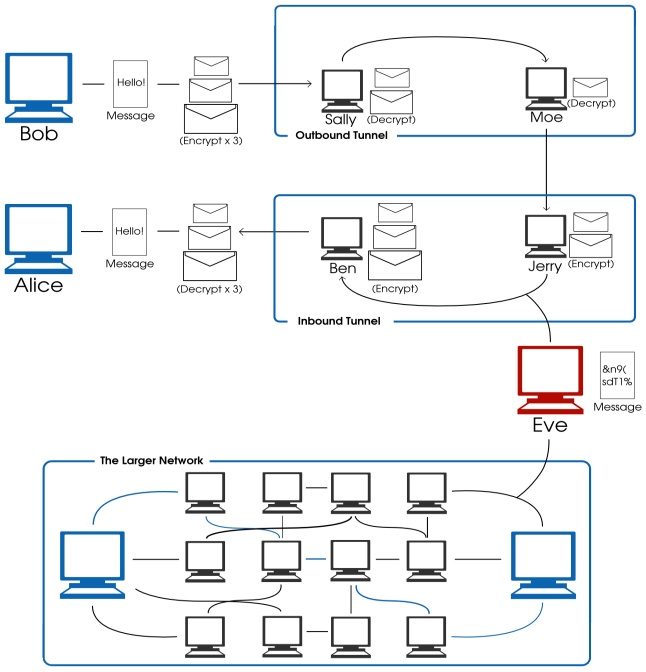

Figure 1. The transfer of messages through the network (adapted from [12]).

2.2 CONNECTING PEERS

To know the flow of messages through the I2P network, we need to know how connections between individuals are established. When Bob wants his message to reach Alice, he needs to know two things: which routers to target in order to send the message through his own outbound tunnel and the tunnel entry points of Alice's inbound tunnels leading to Alice's destination.

These pieces of information are called the router information (*routerInfo)* and a *leaseSet* respectively. The Network Database (netDb) is a distributed hash table based on the Kademlia (see [3]) protocol. It contains this information and consists of a pair of algorithms that share it [2].

For communication Bob first needs to build a tunnel. He uses the routerInfo for this, which he requests in the netDb. The routerInfo contains:

- Routers identity [11];
- Contact address (IP and port number) [11];
- When it was published [11];
- Set of text options (for debugging) [11];
- Signature of the data above [11].

Figure 2 shows how a tunnel is built. Alice gathers info from a list of routers she wants to use for her tunnel. She then sends a build message to the first connection and instructions to give a build message through to the next router, until the tunnel of the desired length has been created.

Now the tunnel has been created, Alice needs information about the destination she wants to reach with her message, in this case Bob. For this she requests Bob's current leaseSet at the netDb.

A leaseSet is a set of multiple leases. A lease contains:

- The tunnel gateway router (by specifying its identity) [11];
- The tunnel ID on that router to send messages with (a 4 byte number) [11];
- Expiry date of that tunnel [11].

Next to these leases, a leaseSet contains:

- The destination itself (a 2048bit ElGamal encryption key, 1024bit DSA signing key and a certificate) [11];
- Additional encryption public key: used for end-to-end encryption of garlic messages [11];
- Additional signing public key: intended for LeaseSet revocation, but is currently unused [11];
- Signature of all the LeaseSet data, to make sure the Destination published the leaseSet [11].

Alice can now send this information through her outbound tunnel, with instructions for the outbound tunnel's endpoint to send her message to Bob's inbound gateway. Bob's inbound gateway sends the message through the inbound tunnel, after which Bob receives and decrypts the message. Alice can also include her own leaseSet with the message, so Bob can contact her for a response, without having to request Alice's leaseSet at the netDb, however this is optional.

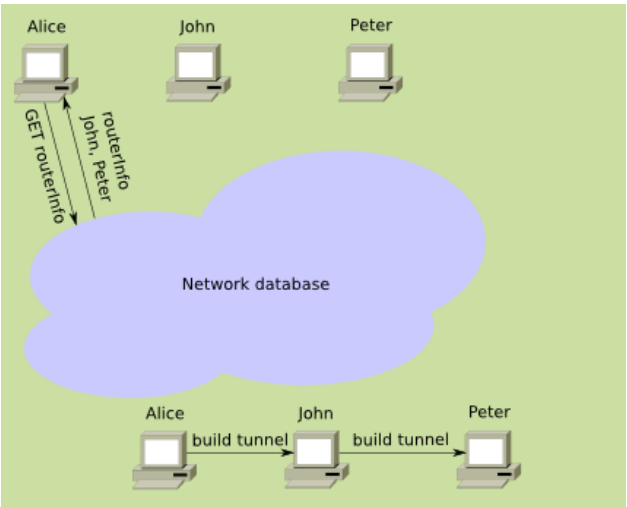

Figure 2. Using routerInfo to build a tunnel [11].

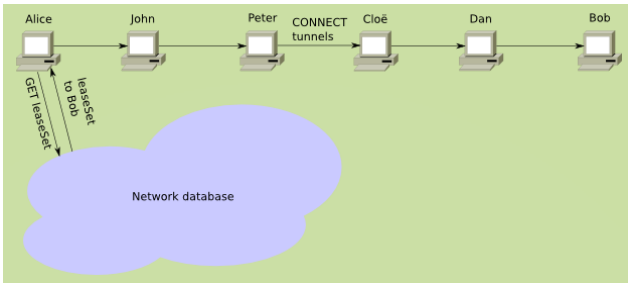

Figure 3. Using a leaseSet to reach a specific destination in the network [11].

2.2 CRYPTOGRAPHY

In order to establish anonymous and secure communication, I2P uses different layers of encryption. Communication between routers is protected by transport layer security. The transport layer encrypts each packet with AES256/CBC. Keys are exchanged by a 2048-bit Diffie-Hellman exchange (see [1]).Tunnel messages are encrypted with AES256/CBC encryption with an explicit IV and verified at the tunnel endpoint with an additional SHA256 hash [7].

Other messages are wrapped in "Garlic messages", which are typical for the I2P encryption. This name is derived from the "Onion encryption" used by another anonymous network application called The Onion Router (Tor). It allows encrypted routing of messages through tunnels. In both Garlic and Onion encryption, a message is wrapped in multiple layers of encryption (like the different layers of an onion covering the center). When the message is sent through a tunnel, one layer is "peeled off" with each hop, revealing the next layer of encryption and instructions to send it to the next router in the tunnel. Garlic encryption additionally offers the possibility of adding multiple messages (cloves) inside the layers of encryption. I2P uses this to send two extra messages next to the original message (figure 4):

- A Delivery Status Message: contains instructions to send a delivery confirmation back to the originator of the message
- A Database Store Message: contains a leaseSet for the destination of the originator of the message

By sending these extra messages, less requests have to be done to the netDb, which improves latency and reduces the risk on traffic analysis attacks [7].

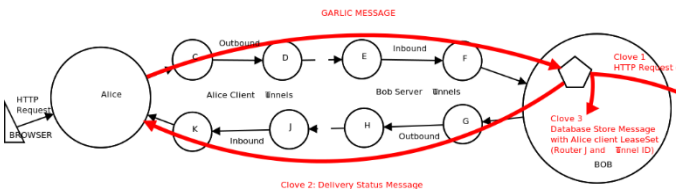

Figure 4. Routing of a garlic message that contains three cloves [7].

3. STRENGTHS AND WEAKNESSES

Multiple applications exist that offer anonymous access to online content. From these, Tor is by far the most widely used and the most important reference point. Therefore, we will compare I2P to Tor in this section before we dive into the strengths and weaknesses of I2P.

3.1 I2P VS TOR

Both Tor and I2P allow anonymous access to online content, make use of a peer-to-peer-like routing structure and operate using layered encryption. But they also differ in some respects.

The most important difference is that Tor was designed as a proxy service to access the regular Internet anonymously, while I2P is designed to create a *network within the internet* and contains all its applications reside within its own borders. Tor has now implemented "Hidden Services" that enable the use of a network within the regular Internet. I2P, on its turn, has implemented an application that allows users to connect to the regular Internet. So while in theory both can be used for the same applications, practice learns that both Tor and I2P are much more efficient when used in the way they were originally intended to be. Another difference is that everyone using the I2P network acts as a 'node' to transfer other people's communication. In the Tor network users need to deliberately choose to become a node. Finally, Tor is written in C whereas I2P is written in Java.

3.2 STRENGTHS OF I2P

- One of the main characteristics I2P is both its principal strength and weakness. Designed to be a self-contained network system, all of its content cannot be reached from outside and vice-versa.
- It can do nearly everything as the regular Internet can do, but anonymously. [12]
- Applications within the network are specifically built for I2P. This it much faster to use them than using Tor's hidden services to access similar applications. [12]
- Short-lived Tunnels in I2P decrease the number of samples that an attacker can use to mount an active attack.
- Peer-to-peer anonymizing service makes sure that all clients on the network also act as routers, which potentially (depending on the size of the user base) offers a high level of anonymity.
- It's very well organized. They have a public and regularly updated task list describing coming developments, a list of supported and a well-documented application programming interfaces (API's) for building applications, a transparent use of the donations, a list of academic papers about I2P and open research questions.

3.3 WEAKNESSES

- There were a few outproxies that one could use to access regular Internet through I2P, but almost all of them are gone. The only outproxy is false.i2p which has [9]. They state if your primary reason to use an anonymous network is to anonymously access sites on the regular Internet, you should use Tor [9].
- The Low amount of users generates a limited number of nodes, which means it is (in theory) less anonymous. In comparison, Tor has a bigger user base, more funding, support of academic and hacker communities and with even developers in the pay list [12].
- It's still considered experimental software, in beta; despite its creation in 2003 [6].
- It's slower than the regular Internet. The encryption and routing within the I2P network adds a substantial amount of overhead and limits bandwidth. However, when more users are connected to the I2P network, its speed increases [9].

4. TYPICAL APPLICATIONS

The applications available through I2P are similar to the regular basic internet applications such as web browsing, e-mail, blogging and forums, website hosting, real-time chat (IRC), peer-to-peer file sharing (Torrents) and decentralized file storage.

4.1. I2P E-MAIL

With a secure and decentralized e-mail service inside I2P, like other applications, it's not possible to send messages to e-mail accounts outside I2P on the regular Internet. I2P uses strong cryptography and mixing techniques to ensure security and anonymity [7]. The default e-mail service is so decentralized that if one would lose a password, it can't be retrieved nor reset; it is lost forever.

4.2. EEPSITE (WEBSITES)

Eepsites are the I2P equivalent of website on the regular Internet. They can contain the same content as regular websites, although for security reasons Javascript and plugins are best disabled. Furthermore, due to the lower bandwidth, most websites employ little bandwidth intensive media such as images or video and very simple styling. An important difference with the regular Internet is the absence of a central Domain Name System (DNS). .i2p Domains can be registered by anyone and are mostly communicated to other users through an address book service [4]. In paragraph six, we'll provide a quick start guide for setting up an eepsite.

5. SURPRISING APPLICATIONS

There aren't any really surprising applications, because the idea of i2p is to have the same features as the regular internet. We found two applications more interesting to use on or with a private and self-contained network.

5.1. ANDROID APPLICATION FOR I2P

Still under development and it does not currently provide strong anonymity [8], but the Android application will enable a mobile device to access the I2P network. We feel it has a lot of potential as a secure and anonymous way to access the network, as a dedicated portable device with prepaid internet sim-cards is more anonymous by default than a PC used for other purposes or connected to a landline. At least it's cheaper and more widely available.

5.2. TORRENTS

Using torrents over I2P is more secure and anonymous than VPN services, and for example not possible in Tor. The torrents available on I2P are mainly copies of the Pirate Bay, backups of leaked government documents, and books that have been banned in some countries [13]. An interesting thing to note is that if an attack or legal measures would ban torrent from the regular Internet, torrents on I2P could gain a lot popularity and support. Again, the main drawback it's the low speed.

6. GETTING STARTED

One of the main services i2p provides is hosting your own website. These hidden services are dubbed eepsites. The i2p client comes with a webserver pre-installed, called Jetty. When you start this webserver, a sample page with all the instructions on to set it up is displayed. This webserver has however limited functionality, as it doesn't support PHP or MySQL for example.

In this quick start guide we'll describe how to set up a webserver with more advanced functionality. Some basic understanding of Apache might come in handy. In this example, we assume a virtual host with the desired capabilities running at mysite.local on port 8383.

6.1 SETTING UP A TUNNEL TO YOUR WEBSERVER

When the service has started and is properly configured, browse to the Hidden service manager:

http://127.0.0.1:7657/i2ptunnel/

Click the Tunnel wizard button, select Server tunnel. Here you select an HTTP tunnel to which you assign a name and a description.

On the Binding address and port screen, enter the hostname and the port (mysite.local and 8383 in our case). Check the box on the final page of the wizard to start the server automatically when you start I2P.

6.2 ASSIGN A .I2P DOMAIN TO YOUR WEBSITE

In the Hidden service manager, click on your newly created tunnel. You'll see that the Website name field is still blank. You can fill in any lowercase address ending on .i2p; mysite.i2p for example. However, beforehand it is wise to check with at least your local address book and a jump service such as stats.i2p if the name isn't already taken. As there is no central authority, such as DNS on the regular web,

conflicting domain names can occur.

While you're still on the settings page of the server, click Add to local address book and save the b32 address or click on details to obtain it. Click Add in the bottom right corner to add your website to the local address book.

6.3 CONFIGURE LOCAL SETTINGS

Find your host file (on Windows, it's located in C:\Windows\System32\drivers\etc) and add the following line to test server:

mysite.i2p=xyz.b32.i2p

Where mysite.i2p is the address you chose and xyz.b32.i2p is the b32 address you saved. When you type in your .i2p address in the browser, your server page should pop up.

6.4 REGISTER YOUR .I2P DOMAIN

Browse to Stats.i2p [4] to add your .i2p site to their database. The local key you enter here is the long key you'll find on the settings page of your newly create tunnel.

It can take a couple of hours up to a few days before your site is fully registered. Remember that only users using the stats.i2p jump service or address book propagation service will be able to find your website at the designated .i2p address. If you want to share your website with others or immediately, share the b32 address (anonymously!).

7. FINAL THOUGHTS

Having all the same features as the regular Internet with the possibility of more anonymity and security is really attractive. We believe that a self-contained anonymous network is a prerequisite for the freedom of speech in the Internet Age. A lot of work needs to be done, not only developing tools but making it more user friendly and quicker. Despite not being as popular as Tor, we believe there is a future for this technology, especially with the use of torrents and e-mailing.

REFERENCES

1. Al-Aali, G., Boneau, B., and Landers, K. Diffie-Hellman Key Exchange. In proc. CSE 331, Data Structures Fall 2000, University of Notre Dame (2000), 67-74.

2. Kubieziel, J. To Be or I2P: An introduction into anonymous communication with I2P. Lecture at 24C3 (2007).

3. Maymounkov, P. and Mazières, D. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. In IPTPS '01, Springer-Verlag (2002), 53-65.

4. Stats.i2p Address Service. http://stats.i2p/i2p/addkey.html

5. The Invisible Internet Project. https://geti2p.net/

6. The Invisible Internet Project: Blog. geti2p.net/nl/blog/

7. The Invisible Internet Project: Cryptography. https://geti2p.net/en/docs/how/cryptography

8. The Invisible Internet Project: Download. https://geti2p.net/en/download

9. The Invisible Internet Project: FAQ. https://geti2p.net/en/faq

10. The Invisible Internet Project: Introduction to I2P. geti2p.net/nl/docs/how/intro

11. The Invisible Internet Project: Technical Introduction. https://geti2p.net/en/docs/how/tech-intro

12. The Invisible Internet Project: Tor / Onion Routing. https://geti2p.net/en/comparison/tor

13. The Tin Hat: I2P: Welcome To The Darknet | How to Configure I2P. https://thetinhat.com/tutorials/darknets/i2p.html

14. Timpanaro, J.P., Chrisment, I., and Festor, O. I2P's usage characterization. In proc. TMA'12, Springer-Verlag (2012), 48-51