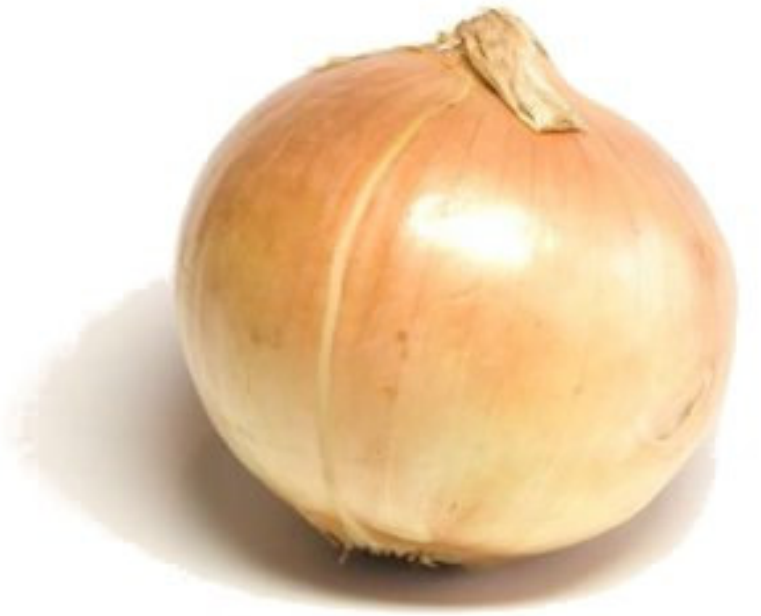
Anonymous Networks



Onion Routing with TOR, Garlic Routing with I2P

Henrik Erkkonen
Jonas Larsson
Datateknik, Chalmers
EDA390, Computer communication and distributed systems

Introduction

Onion routing is a technology aiming to provide anonymous communication between entities on a network. The goal is to provide low latency connections transparent to the end user, while the information exchange still is resistant against traffic analysis and other attacks. This is achieved by a set of encrypted layers and frequently changing paths between a subset of the routers that participates in the routing system.

The concepts onion routing was introduced by David Goldschlag, Michael Reed, and Paul Syverson. Their idea is partly built on “mix networks”, introduced by David Chaum. Syverson later co-started the Tor network, which is the most used onion routing system by the general public today. Technology for anonymous communication is controversial. Some people think that the possibility to hide their identity online is plain bad, while others see it as a human right.

Garlic routing is an evolution of onion routing with changes in how messages are wrapped and routes are chosen. I2P, The Invisible Internet Project is an anonymous peer-to-peer network that uses garlic routing and was developed independently and parallel to Tor. There are a lot of differences in how the network is organized compared to Tor, and a lot of similarities.

Motivation for anonymity

The most common argument against tools for anonymous communication is that criminals can use it to plan future crimes, exchange illegal content etc. without revealing themselves. The people behind Tor says that while this is true, it should not be a valid reason to why ordinary citizens should not be able to communicate anonymously. This is because criminals already have means to be anonymous. Criminal activities can for example be planned using encrypted end-to-end communication between public computers at places such as libraries. Other possibilities are the use of stolen cell phones, or computers hijacked by trojans and other malware. Nowadays identity theft is becoming more and more common among people with dark intentions. So the conclusion is that criminals already have means to hide their identity while ordinary people don’t. Surely criminal minded jerks can use Tor to get away with their actions, but they already have better and more effective means to do so.

So which legitimate reasons for ordinary people to be anonymous are there? Common reasons to use Tor are to avoid being tracked by advertising companies on the Web, reach Internet services and sites blocked by the ISP or participating in chat rooms for victims of all kinds of abuse. Most people can probably think of at least one reason to be anonymous on the net without causing anybody else any harm. Government agencies use Tor for intelligence gathering and people in China and other countries without freedom of speech use it to communicate with other freedom seekers.

Tor – The Onion Router

Overview

Tor is a distributed, anonymous network. The network is not run but a certain organization, but by a diverse set of organizations and individual donating their bandwidth and processing power. The software is open source, so anybody can check for backdoors and other flaws. The project is maintained by The Free Haven Project, and its web resources are donated by the Electronic Frontier Foundation.

The routing in Tor is done on the transport level in the protocol stack, and only supports TCP. Applications access the network through the SOCKS interface, which means that all applications with support for SOCKS can use Tor for anonymous communication, without needing modification. The network consists of Tor nodes (routers), run by contributors, and central directory servers run by the maintainer. The directory servers are a database of all routers which both routers and Tor clients use to gain knowledge of the network. A few directory servers have the risk of single point of failure, so most routers pass the directory database around amongst the peers in the network for back up reasons. This is also done to lower the load on the main directory. A list of some directory servers is distributed with Tor to facilitate joining the network (bootstrapping).

The directory servers are in fact a group of established routers that monitor the network to build a view of the entire topology. All others can fetch lists of routers and routers can submit their information. There the entries are cryptographically protected with signatures and only information from approved routers will be published in the database, to avoid attacks where someone adds a lot of subverted nodes. There is no automatic system to approve routers; the directory server administrators do this manually.

Each node has a long term identity key and a short time onion routing key. The identity key is for signing of TLS certificates and the description of the nodes capabilities. The onion routing key is used in communication with other nodes to decrypt signaling protocol messages and to negotiate session keys.

Tor is almost always used together with a local intermediate proxy server such as Privoxy. This is because many Internet applications send data that can be used to help gather information about a user. A common example is the browser brand and version and operating system. This is not a flaw in Tor itself, but more of a design choice; Tor is not designed to provide anonymity on the application level. There is also a privacy problem with DNS lookups, which often are sent outside the intermediate proxy through the regular network, which can expose what services a users is connecting to through Tor. Tor does not conceal that a user is connected to Tor, but it hides what the user does on the network. This is usually referred to as that Tor is a non-steganographic network.

Traffic and routing

Circuit set up

The circuit is built from the entry point (user) one step at a time. A circuit ID is chosen randomly, and a Diffie-Hellman key exchange is initiated. When done, the starting point has negotiated a symmetric session key with the first hop. The entry point sends a request to the first hop to extend the circuit, containing the new node. The Diffie-Hellman process is repeated, but all messages to node number two are relayed through the first hop. In every step the messages is encrypted with the negotiated session keys, or when not already negotiated, the receiving hosts onion key.

Traffic through the Tor network

When the circuit is set up, it is used to relay data. The last router in the path is called the exit node. The data to be sent is encrypted in several layers, like an onion (hence the name Onion Routing), together with routing information with the data destined for the exit node at the core of the onion. This core is then encrypted for the router closest to the exit, along with information of which the exit router is. This procedure is repeated for all other routers in the path. When the sent packet reaches the first router in the path, the router decrypts the routing information to the next hop, and the encrypted data for that hop. This process is repeated until the packet reaches the exit node. At each intermediate step, the current router is unable to see where the data is destined to, where it is originating from or the data itself. The data is sent in clear between the exit node and the ultimate destination, but the exit node has no means to know where it's from.

Tor carries out integrity checking on the data send through the network. This is done to prevent malicious nodes to corrupt data. In early versions of onion routing this was a potential problem; an adversary to the network could install a high performance node that corrupted the data sent through it. A mechanism built on cryptographic hash functions is used at the end points to detect this, and another circuit can be built.

Tor has a signaling scheme for use inside the circuit, which can be used to change exit nodes. This makes it virtually impossible for observers to track changes in the circuit. Such information, if available, could be used for attacks on the network. The signaling scheme is also used for congestion control. The congestion control is done on an end to end basis, much like TCP but with anonymity features added.

Hidden Services

An important feature of Tor is the Hidden Services. This feature allows any user to set up an Internet service, such as a web page or a message board, and let anybody use it without knowing where it is located or who is behind it. It also works the other way around; the service operator has no knowledge of who are using the service. Hidden services have the top level domain .onion, and the host name has to be looked up using the Tor network. This can be a problem, because all applications do not currently forward DNS lookups via SOCKS. Tor Hidden Services can be accessed by all users of the Tor network, and are designed to resist censorship, DDoS and physical attacks respectively. But how can a server everybody can access possibly hide both

it's location on the network and physical location? To find and reach Hidden Services, Tor uses a concept called "rendezvous points".

To set up a hidden service such as a web server, the owner generates a public/private key and selects a number of onion routers (called introduction points) to which tunnels are set up and the service is announced together with the public key on a Service Lookup Server. The service is then announced to users by some means (e.g. message board). When a user wants to access this service they find one of the introduction points through the Service Lookup Server. The user also chooses a router as "rendezvous point" to which a tunnel is set up. For the actual connection, the introduction point is told about the rendezvous point, which it forwards to the service owner. The service owner then sets up a tunnel to the rendezvous point and a connection between user and service is complete. All these exchanges are protected with public key cryptographic methods. Even if the primary use of hidden services is anonymity, it can also be used to operate a server from inside a firewall.

Abuse and attacks

Abuse

Running an exit node is associated with a risk for abuse. There has been a lot of misuse of Tor for spam, threats, hacking, abusive IM and illegal file sharing. After the introduction of more strict policies for exiting traffic this has been reduced. The default exit policy now rejects private IP subnets, email (SMTP), Usenet News (NNTP), Windows file sharing and also a number of popular file sharing applications (Kazaa, eDonkey, Gnutella, Napster, Bittorrent).

However, an operator running an exit node can still expect to be contacted by someone being abused through Tor. There might also be issues with ISP agreements forbidding traffic relaying. The Electronic Frontier Foundation has prepared answers to DCMA Cease and Desist complaints for users to give their ISP's if copyright infringement has occurred through a Tor exit node, and the first conviction of a Tor router operator is yet to be seen. If a user is willing to cope with the risks of having to talk friendly with their ISP, running a Tor node is easy. The software is downloaded in a convenient package, and is easily installed. If not behind a firewall or NAT, almost no configuration is required; otherwise ports must be opened or forwarded. Optional configuration of bandwidth usage can also be done.

Attacks on Tor

The Tor network does not provide bullet proof anonymity and confidentiality, but the level provided is considered to be sufficient considering its purpose. Traffic from exit nodes to the final destination is sent in plain text to the destination, meaning that the traffic can be intercepted by any entity located in between, including the exit node. This is not a threat to anonymity, but could be to confidentiality. The solution is to use encryption on the application level between the communication parties. If both parties run Tor servers, the exit node is chosen as the destination node, and the traffic is encrypted all the way. Plain text snooping is not an attack on the system, but more of a part of its design. Many users are unaware of it though.

There are some attacks that can reveal the identity of a Tor user to some degree, the main one being timing analysis excluding the DNS problem mentioned earlier. By watching packets leaving a user and entering a target server one can correlate the traffic and make probable that the user is in fact connecting to it. This however requires the possibility to monitor both user and target, and is not practical for most individuals and organizations.

Tunnels in Tor are reused by different applications so by observing the traffic at an exit node one can correlate different traffic streams, which may give more information about a user. E.g. if plain text chat data and a file sharing protocol exits the same node with these two can be assumed to be correlated with a certain probability (higher if the exit node has low traffic volume).

Another probability-based attack is the intersection attacks, where an adversary observes when dynamic routers leave the network. This breaks some connections and by looking at traffic surviving this one can minimize the number of probable paths.

To increase probability of some of the attacks mentioned or just hurt the network one can flood the routers with requests. This may cause a denial of service, as the mass of encrypted packets requires significant computing resources to process or simply exhaust the available bandwidth.

One attack on anonymity on web surfers using Tor can be carried out with Java applets, ActiveX controls or any other programs that run in virtual machines. These programs can collect local information and send it back the website owner. Disabling such features can circumvent the attack, but may decrease the browsing experience or even render some sites non-functioning.

I2P and Garlic routing

Overview

Garlic routing is based on onion routing with the following major change: Onion routers have the possibility to join several messages with independent routing information on each level into a new onion for the next node. The messages ("cloves", hence the name garlic) in an onion message can have arbitrary options such as a request to delay the message in the next node for some time or end there, while the rest of the clove is disassembled and reassembled in new onions. The onions can also include padding to masquerade how many actual cloves there are. All these operations make traffic analysis much more difficult as long as there are enough messages.

I2P, The Invisible Internet Project, was started in 2003 with the purpose of enabling anonymous communication in a dynamic decentralized network resilient to attacks. All communication is end-to-end encrypted and implemented as a garlic routing network layer leaving it open for use by any kind of client-server or peer-to-peer using it.

The I2P developers are anonymous to the general public and only known by their pseudonyms; the founder and main developer calls him self "jrandom". The project is still in an alpha stage and is not considered mature for broad use yet.

I2P

There are some major differences between Tor and I2P. First of all I2P is a transport protocol comparable to IP. Data is sent in packets/datagrams and while Tor clients randomly determine a tunnel path for a connection, I2P has another way: In I2P the tunnels are one way. Each node has a number of outgoing and ingoing tunnels to different peers. When a single packet is to be sent, it is addressed to one of the receivers ingoing tunnel endpoints (found through the network) and sent on one of the own outgoing tunnels. This way the sender has no information about the path after the outgoing tunnels endpoint. As this is done on the packet level instead of connection, it has some reliability advantages, as a node that disappears will not disrupt the connection and the connection bandwidth will be distributed for higher throughput. The service is unordered and best-effort, but supports a streaming service in the same way that TCP uses IP.

Another difference is the distribution of the network. While Tor has a concept of central directory servers, I2P is fully distributed requiring a bootstrapping operation to find one peer to be able to join the network. There is a file with some known nodes published on the I2P development web page. Once one router has been found it can be queried for more known routers.

Each node keeps private statistics on latency and behavior for the known routers. This

is used to place the router in one of four categories: fast and high capacity, high capacity, not failing, and failing. When routes are set up for outgoing messages routers are used in falling order. Routers from the lower categories are used to explore the network to try to find other alternative paths. The algorithms for peer selection have been changing between releases and are expected to be changed again.

I2P was not designed to reach regular Internet services anonymously, and there are no exit nodes in the protocol. As detailed in the Tor part the exit nodes are susceptible to abuse and there are some security issues weakening the anonymity of the users. However HTTP proxies have been developed which allows traffic to exit I2P and reach normal web pages, but the node running the proxy has to be known. Regular TCP/IP applications can not be used directly, but have to be modified or run through software known as I2PTunnel to connect to other I2P hosts. There a number of I2P applications, among the most used are I2Phex (file sharing based on Gnutella) and I2PSnark (a BitTorrent client).

I2P has had the concept of hidden services from the beginning using a setup that is similar to the one Tor adopted. The TLD used is called .i2p and the web pages are called Eepsites.

Attacks

On the application layer I2P is vulnerable to the same anonymity attacks as Tor (and other similar anonymous services such as Freenet): applications/protocols reporting their real IP and leaking local information through JavaScript/Java etc.

Another possible anonymity attack against I2P that has been discussed is flooding a specific node with requests effectively causing a denial of service and monitoring if a hidden service goes offline. By repeating this from different locations and using statistical analysis one can infer that the service was hosted on that node (false positives may be the closest node relaying traffic to that service). This becomes increasingly difficult as the network grows.

Most of the statistical attacks from Tor apply to I2P, such as observing timing of packets leaving nodes and which peers connect to which. The garlic part makes these attacks much harder, but not impossible if the attacker controls a lot of nodes near the user. Because I2P is still in early development there may also be flaws in the actual implementation that can reveal information about the user or be used for denial of service.

A number of different scenarios for denial of service attacks exist, from the most simple ones where nodes are flooded with traffic which either consumes all bandwidth or CPU resources for cryptographic operations.

Statistics

The authors of Tor estimate that there are 450 running routers (Mar 2006) and it has been growing exponentially in the last 24 months. Measured bandwidth capacity is approximately 100 MB/s and utilization is around 50 MB/s on average.

For I2P the last reported number was 300+ known running routers (Apr 2006) although approximately 1000 unique peers has been seen on the network (Oct 2005). There are no reports on available bandwidth as I2P does not have directory servers collecting statistics as Tor has.

Summary & Conclusions

Anonymous network services is a large field of research and development that steadily continues to grow. The interests and demands of the general public are increasing all the time. This can easily be seen when looking at the number of related projects which are being actively developed. The source for this demand may be all the lawsuits against users of file sharing applications, and in some countries, the outlawing of such applications. But not only individuals with a liking for file sharing want anonymous networks. Human rights activists fight for the right of freedom of speech without unmotivated eavesdropping by the government and journalists want to protect their sources. In many western countries, laws permitting such eavesdropping or wiretapping by government authorities is already in place, or will soon be.

Equally much as the topic of anonymous communication is controversial, it also shows the creativity of a group of dedicated individuals to fight back against laws and authorities by the means of technology. It is therefore ironic that Onion Routing originates from the US Navy Research lab. Most anonymous systems is however created and developed by individuals or academic institutions.

Tor and I2P are two systems we believe will still be in use in the following years. They have proven to be working and are constantly gaining popularity. The interested reader should also look up the following systems, which also serve as an example of how rich the field of anonymous networks really is:

Share: A closed source file sharing application from Japan. Developed by an anonymous engineer, because file sharing applications are illegal in Japan. It implements a large virtual distributed hard drive.

FreeNet: One of the most used anonymous networks. In difference to Tor, it does not allow anonymous communication with the network outside FreeNet. Instead, all users contribute with their own disk space to store encrypted information for others to access. A similar system is called Entropy.

GNUnet: A system that supports direct downloading of files through anonymous tunnels similar to those used in Tor, but also supports diffusion of data as in Share and

FreeNet. This means that all shared information is spread out onto potentially all users' hard drives, hiding the original source of the information.

Sources

Roger Dingledine, Nick Mathewson, Paul Syverson. Tor: The Second-Generation Onion Router. Usenix Security 2004, August 2004.

Roger Dingledine. The Free Haven Project - Design and Deployment of an Anonymous
Secure Data Haven. MIT Master's Thesis, June 2000.

Number of Running Tor routers.
http://www.noreply.org/tor-running-routers/

Wikipedia on I2P
http://en.wikipedia.org/wiki/I2P

Official I2P site on Garlic Routing
http://www.i2p.net/how_garlicrouting

jrandom (Pseudonym)
Introducing I2P: A scalable framework for anonymous communication
http://dev.i2p.net/cgi-bin/cvsweb.cgi/i2p/router/doc/techintro.html?rev=HEAD