Proceedings of Informing Science & IT Education Conference (InSITE) 2014

Cite as: Grahn, K. J., Forss, T., & Pulkkis, G. (2014). Anonymous communication on the internet. *Proceedings of Informing Science & IT Education Conference (InSITE) 2014* (pp. 103-120). Retrieved from http://Proceedings.InformingScience.org/InSITE2014/InSITE14p103-120Grahn0483.pdf

Anonymous Communication on the Internet

Kaj J. Grahn, Thomas Forss, and Göran Pulkkis
Arcada University of Applied Sciences, Helsinki, Finland

kaj.grahn@arcada.fi thomas.forss@arcada.fi goran.pulkkis@arcada.fi

Abstract

There are many kinds of systems developed for anonymous communication on the internet. We survey a number of systems and evaluate their security. Among these systems we compare functionalities like Onion Routing, anonymous VPN services, probabilistic anonymity, and deterministic anonymity. Other types of anonymous communication such as messaging, peer-to-peer communication, web use, emailing, and use of other Internet applications are also presented. We follow up by presenting different types of attacks with the purpose of identifying anonymously communicating users. These attacks fall into the following categories: internal/external attacks, passive/active attacks, and static/adaptive attacks. We describe the following attacks as well as known protections against these attacks: predecessor attacks, intersection attacks, timing attacks, and Sybil attacks. Lastly we discuss design choices, operation, and security of the current TOR network – The 2G Onion Router. Access control methods to restrict malicious use of TOR are also proposed. In conclusions the significance of anonymous communication is outlined.

Keywords: anonymity, privacy, onion routing, proxy, relay, security, security attack, TOR network.

Introduction

In an age when much information that is sent over the Internet is stored and analyzed by different companies (Joshi, 2009), governments (Burke, 2013), and net criminals (Gercke, 2012), it has become more and more important for people and entities to be able to anonymously send and receive data over the Internet. It has also become important for people and entities to be able to hide their whereabouts. By knowing the sender's or the receiver's original IP address a hostile entity can find the geographic location or even the real identity of a person behind the communication (Palme & Berglund, 2004).

This paper first presents the principles of anonymous communication. Then the current anonymous communication solutions such as onion routing, the Anonymizer service, anonymous messaging, anonymous P2P communication, and anonymous use of Internet applications are surveyed. After this follow an analysis of security attacks on anonymous communication and a presentation of known defenses against these attacks. Finally, the currently most frequently used anonymous communication network, The 2G Onion Router – usually called the TOR network – is thoroughly described. Access control methods to restrict malicious use of TOR are also proposed.

Anonymous Communication

Anonymous communication was first concretized by Chaum's (1981) proposal for protection of communicating parties with so called mix networks. The principle of a mix network is to use proxy relaying servers and encryption for anonymity of senders and receivers. A layered encryption is created by encrypting data communication in each proxy using public key cryptography. Also a basic idea of anonymous email communication is presented. The network nodes accept public key encrypted emails, decrypt and send them on. Timing alteration, which means delaying forwarding messages at some point in the network, is also carried out in order to make path tracing harder.

Anonymity and privacy are closely interrelated concepts. For anonymous mobile communication location privacy is a necessity. In Liang, Li, Li, and Deng (2012) location privacy means that the identities of communication parties, the time of the communication event, the physical locations of communication endpoints, and the context of communication must not be disclosed to third parties.

K-anonymity is a concept presented in Gruteser and Grunwald (2003). K-anonymity means that a network user cannot in practice be distinguished from at least K-1 other network users when K is sufficiently large. K-anonymity implementations are described in Gedic and Liu (2008) and classified as deterministic anonymity in Kesdogan, Egner, and Büschkes (1998), where also the concept probabilistic anonymity is introduced. The protocol Stop-and-Go-MIX (SG-MIX) is proposed for providing probabilistically secure user anonymity without identity verification.

Anonymous Communication Solutions

In Marques and Zuquete (2011), four current solutions to anonymous communication are presented: Onion Routing, Crowds, Freenet, and Anonymizer. Solutions for anonymous communication as well as anonymity solutions for messaging, peer-to-peer communication, web surfing, sending and receiving email, and other Internet applications are also presented in this chapter.

Onion Routing

Onion routing provides anonymous connections that are strongly resistant to both eavesdropping and traffic analysis (Reed, Syverson, & Goldschlag, 1998). The procedure starts by connecting the client, for example an application, to an application proxy (SOCKS) as shown in Figure 1. Protocol-specific connections from the applications are accepted and converted into a generic protocol (MLSA, 2005).The packet is then passed on to an onion proxy, where a special data structure, an onion, is created. The onion is multiply encrypted and finally sent to an entry funnel. Repeatedly encrypted messages are sent along an unpredictable route through several network nodes called onion routers. These routers communicate with each other through TCP tunnels.Traffic passes bi-directionally along those circuits with minimal latency. A layer of encryption is removed by each onion router and the message is sent further to the next router. This procedure prevents intermediary nodes from knowing the origin, destination, and contents.

Onion Routing provides an anonymous real time virtual socket connection through a proxy server. The Onion Routing architecture can be easily used by a number of applications because many protocols are adapted to work with proxy servers (Goldschlag, Reed, & Syverson, 1996).

We follow the description given in MLSA (2005). A network node is defined by a node number S=1…N, a public key Su, a private key Sr, an encryption function E[key](data) and a decryption function D[key](data). Data encrypted with a public key Su can be decrypted with the corresponding private key Sr, and vice versa:

D[Su](E[Sr](data)) = data.

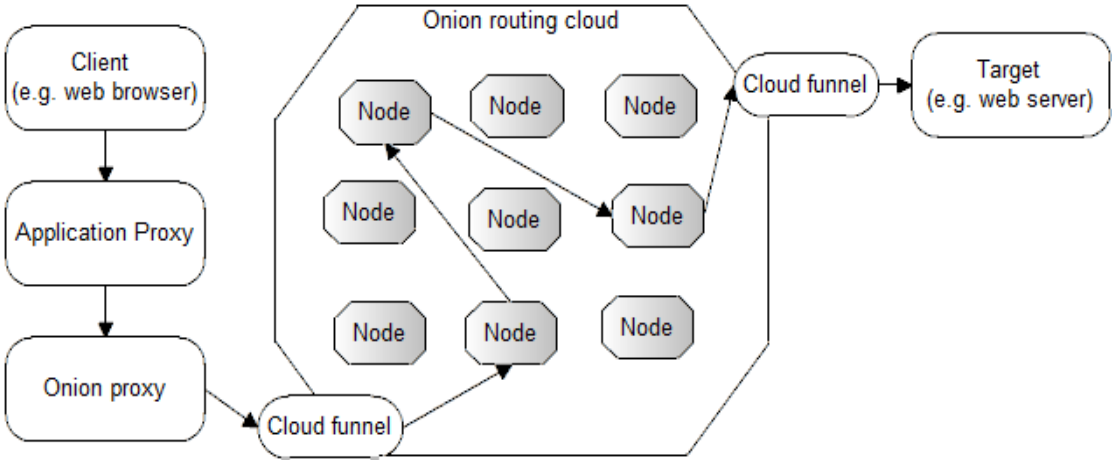

Figure 1: Onion routing cloud with example path.

We further assume that the random sequence of routers constructed by the proxy is <4, 3, 5>. In this sequence the first router is an entry funnel and the last an exit funnel. The data packet, called an onion and sent to the entry funnel by the onion proxy, will then have the following structure:

E[4u](3's IP address, E[3u](5' s IP address, E[5u](data)))

The entry funnel, router 4, decrypts the onion with its private key revealing router 3's IP address and a chunk of encrypted data as seen in step I in Figure 2. The process repeats itself and a virtual circuit is established to the exit funnel, as seen in steps II - IV in Figure 2. Thus the circuit doesn't need to include routing information. When a response message enters the exit funnel, data is encrypted with the private key of the funnel. The same procedure continues at every circuit router until the data receives to the onion proxy where the message is recovered as

D[4u](D[3u](D[5u](encrypted onion)).

(MLSA, 2005)

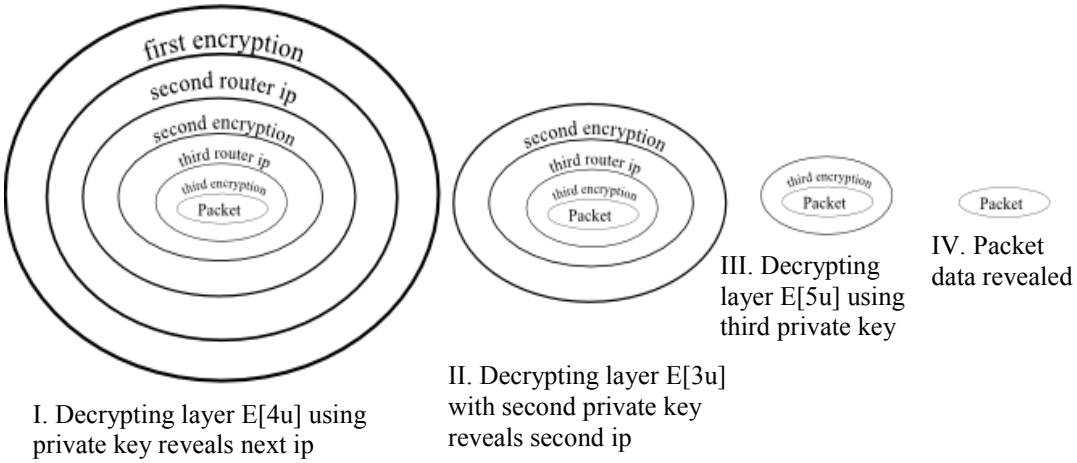

Figure 2: Onion with triple encryption.

A message sent in an onion router architecture contains a virtual circuit identifier, a command (*create*, *destroy*, and *data*), and data. The onion occurs as the data field. A node receiving a *create* command along with an onion sends another *create* along with a virtual circuit identifier and the onion to the next node (Goldschlag et al., 1996). If a node receives another command than *create*, *destroy* or *data* a *destroy* command is sent back through the virtual circuit. The role of the com-

mand *data* is to pass a stream of data from the initiator along the virtual circuit together with other control information.

Anonymizer

Anonymizer is a commercial VPN service, which makes Internet activity untraceable (Anonymous, 2013). This design is a single point system, i.e. requests for web pages go through a single website, and the proxy usually offers an encrypted communication channel for traffic back to the user. In a networked system like Onion Routing, requests are sent through multiple layers of anonymization. A single-point system offers less resistance to sophisticated traffic analysis than a networked design (How, 2011).

The three basic components of the single-point system are (Ling, Fu, Jia, Yu, Xuan, & Luo, 2012):

- *Anonymizer Client.* Commercial software which is run by the client to anonymize the data.
- *Anonymizer Server*. It consists of one reverse proxy/Network Address Translation (NAT) server, several SSH (Secure Shell) servers and web proxies. The cluster of SSH servers and web proxy servers is used for load balancing. The encrypted client TCP traffic of POP3, SMTP, FTP and HTTP is dispatched to an SSH server via an SSH tunnel. The traffic is then decrypted and forwarded to a web proxy.
- *Destination Server.* TCP applications are run by this server.

See Figure 3.

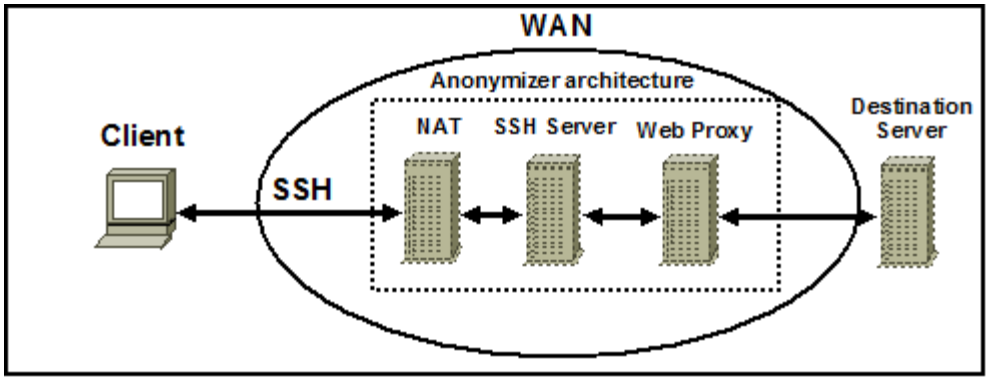

Figure 3: Architecture of the Anonymizer network.

In Ling et al. (2012), it was shown that the size of HTTP packets in the Anonymizer network is very dynamic and random at the client. The work exploited the Anonymizer architecture and showed that packet size based covert channel attacks degrade the anonymity.

Anonymous Messaging

Dining Philosophers Network (DC-Net) is an anonymous messaging protocol proposed by Chaum (1988). More scalable anonymous communication protocols, for example Herbivore (Goel, Robson, Polte, & Sirer, 2003), are based on DC-Net. A formal specification of DC-Net is published in Goel et al. (2003).

A group messaging protocol, Dissent, providing provable anonymity, is proposed in Corrigan-Gibbs and Ford (2010). Dissent is demonstrated for groups of 40+ members with a prototype implementation programmed in Python and using OpenSSL public key cryptographic functions.

Anonymous P2P Communication

Freenet (Clarke, Sandberg, Wiley, & Hong, 2001) is like a social network, in which files can be anonymously shared, Web pages accessible only through Freenet can be anonymously published and browsed, and anonymous chat forums are available. Freenet is implemented as a decentralized peer-to-peer network in order to decrease vulnerability to security attacks.

I2P is a middleware layer implementing an application framework, which is built around a network host executing a software component called I2P router and providing connectivity for local I2P applications. An application either hosts a service as a server or accesses services as a client. Service applications include web sites, file sharing, email, and chat. An I2P router maps connections to client and service applications to packet based I2P tunnels, which provide anonymity with the use of onion routing. A decentralized p2p network running as an overlay upon IP implements connectivity between applications, which use either the TCP based protocol NTCP (New I/O based TCP) or the UDP based protocol SSU (Secure Semireliable UDP). All metadata is stored in a Distributed Hash Table (DHT), which is called netDB and ensures network scalability. I2P provides a separate network called Darknet to client and service applications. Within a Darknet all end-to-end connections are encrypted. Sybil attacks (Douceur, 2002) and other security attack types can be combined to uncover anonymity of I2P users. Ongoing research is focused on improving defenses against known attack types. In February 2013 an operational I2P Darknet on the Internet had about 20000 users (I2P, 2013; Egger, Schlumberger, Kruegel, & Vigna, 2013).

A scalable and decentralized p2p anonymous network layer, Tarzan, is presented in Freedman, Sit, Cates, and Morris (2002). Other proposals for anonymous p2p communication are AP3 (Mislove, Oberoi, Post, Reis, Druschel, & Wallah, 2004), Salsa (Nambiar & Wright, 2006), and MorphMix (Rennhard & Plattner, 2002). Mute (Mute, 2013) is a p2p network for anonymous file sharing.

GNUnet is a framework for secure p2p communication (GNU's Framework, 2013). GNUnet's Anonymity Protocol (GAP) is described in (Bennett & Grothoff, 2003).

Anonymous Use of Internet Applications

SmartHide (2013) is a commercial anonymizer for Internet applications. SmartHide hides the source address and protects communication with strong encryption.

Java Anon Proxy (JAP) is an anonymous proxy service for Web surfing (JAP, 2011). A user connects to a web service with encryption through a sequence of intermediaries called mixes. As default, a predetermined sequence of linked mixes is available, but a user can also choose between different sequences of linked mixes.

Crowds (Reiter & Rubin, 1998) is an anonymity system preventing a Web server from learning anything about identification of browsing clients, their IP addresses and domain names.

Hordes is a proposed protocol for anonymity of the initiator of network communication. Hordes uses multicast routing to achieve anonymous reception of data. Hordes has been shown to provide the same degree of anonymity as Crowds (Reiter & Rubin, 1998) and Onion Routing (Reed & al., 1998). Several performance advantages can be found (Levine & Shields, 2002).

GTunnel (Garden Networks, 2013) is a freeware Windows application to be used as a local proxy of a Web browser or some other Internet application. GTunnel provides source anonymity and protects content with strong encryption.

Your Freedom is a commercial service for anonymous use of most Internet applications (Your Freedom, 2013). A user installs the client application, which is available for all common operating system platforms. Network applications on the user's computer can then be configured to use

the Internet through the anonymous Web proxy or the anonymous SOCKS proxy created by executing the installed Your Freedom client.

SubRosa is an emulator of the anonymous communication network TOR (Tor, 2013). SubRosa runs on the global research network PlanetLab (PLANETLAB, 2007) and has been designed to collect timing data for defense algorithms against timing attacks (Levine, Reiter, Wang, & Wright, 2004) on anonymous communication systems. (Daginawala & Wright, 2008)

Proposals for anonymous emailing are Babel (Gülcü & Tsudik, 1996), Mixminion (Danezis, Dingledine, & Mathewson, 2003), and Mixmaster (Mixmaster, 2012). Mixminion is also proposed to be used for securing financial communication against traffic analysis attacks (Mathewson & Dingledine, 2004).

Security of Anonymous Communication

The purpose of attacks against anonymous communicating users is to identify the sender and/or receiver. Security of the content of data packets is orthogonal to anonymity and security of the route protocol (Zhu, Wan, Kankanhalli, Bao, & Deng, 2004). Attacks against anonymous and secure routing in ad hoc networks are usually classified into passive and active attacks. A *passive attack* is one in which the intruder eavesdrops or silently refuses to execute the function requested. The attacker does not modify the message stream in any way nor disrupt the operation of the routing protocol. Such an attack is difficult to detect. The purpose of an *active attack* is to degrade or prevent data flow between nodes. Such actions may be replication, modification, and deletion of exchanged data. Passive and active attacks may be launched at the same time and thus amplify the final result of attacking. An adversary may first identify the route and then launch DOS (Denial of Service) or DDOS (Distributed Denial of Service) attacks.

A more systematic listing of attack types is given in (Raymond, 2000):

- **Internal/external.** Attackers can be distinguished on whether they are participants in the network or not.
- **Passive/active.** Attackers can actively change the status of the network or remain passive.
- **Static/adaptive.** Attackers can't change their resources once the attack has started or they can continue to build up their capabilities.

According to Marques and Zuquete (2011) the most common security attacks against anonymous communication systems are predecessor attacks, intersection attacks, timing attacks, and Sybil attacks.

Predecessor Attacks

A predecessor attack is a passive attack, very difficult to detect, in which an initiator of an anonymous connection can with some probability be identified from information about predecessor's proxy paths (Reiter & Rubin, 1998). An initiator, who communicates anonymously with the same responder across path reformations, can be revealed by an attacker logging any message sending node in each path reformation. Upper bounds on the time required to disclose the initiator of anonymous communication have been derived for Onion Routing, Crowds, DC-net, and SG-MIX. Defense against predecessor attacks is in Onion Routing based on static path selection. For example, an attack fails when the first node in a path is a trusted node (Sylverson, Tsudik, Reed, & Landwehr, 2000). For DC-net the time to disclose anonymity with a predecessor attack can be considerably prolonged by adding a few edges to the chosen topology or by random choice of topology. For anonymous p2p communication effective defenses against predecessor attacks are still not known (Wright, Adler, Levine, & Shields, 2004).

Intersection Attacks

An attacker who knows which communication parties are active at any given time can make observations to determine which parties communicate with each other. In this attack it is assumed that a network user is typically in contact with a relative small number of other communicating parties (Danezis & Serjantov, 2005). Resistance against intersection attacks is obtained by the use of an extended anonymous communication scheme called Buddies architecture (Wolinsky, Syta, & Ford, 2013). In this communication architecture communication parties are dynamically grouped in buddy sets, for which communication is controlled in order to make buddies in a set indistinguishable from each other when network traffic is analyzed.

Timing Attacks

An attacker analyses the timing of messages sent to an anonymous communication system in order to find correlations. Protection against timing attacks is obtained by combining rate cover traffic along the length of the entire communication path with sufficiently frequent random dropping of some cover traffic (Levine et al., 2004).

We assume that the attacker has access to a particular set of network mixes. A mix hides the relationship between the incoming and outgoing packets by using various techniques such as encryption, delays and traffic covering. The attacker wants to determine the mixes that form the communication path. We mention three timing analysis attacks and corresponding defenses. These can be found in (Gianchandani, 2012). The timing attack types are:

- **Watermarking**. One technique is to transparently watermark the packet flow by slightly adjusting the timing of selected packets (Wang, Chen, & Jajodia, 2007). If the flow after transformations can be uniquely identified it can be linked to the original sender and receiver.
- **Flow correlation attack**. An adversary analyzes the network traffic and attempts to correlate the traffic flow over an input link at a mix with the traffic flow over an output link. Time domain and frequency domain methods are often used. Path reconstruction is done on a mix-by-mix basis. The effectiveness of the attacks is measured in terms of the probability that the receiver is correctly recognized.
- **Selective cross correlation attack**. The incoming and outgoing traffic streams are divided into non-overlapping windows. The packets in the incoming window are then compared with the corresponding packets in the outgoing window. If there is more traffic in the outgoing window we can conclude that dummy traffic has been added. All windows where we have cover traffic are removed and cross correlation analysis is performed over the other windows.

Defenses against timing analysis attacks are

- **Adaptive padding defense**. Arbitrary packets are inserted into statistically unlikely gaps in the packet flow (Shmatikov & Wang, 2006). In this way, timing correlation is reduced without adding any latency to the traffic. When the user's traffic rate is low, the mixes increase the padding rate. The method prevents the attacker from determining which of the multiple simultaneous connections is being carried on a given network link.
- **Defensive dropping**. The client creates dummy packets and marks them to be dropped at any intermediate mix at random (Daginawala & Wright, 2008). The other nodes of the path are not aware of the dropped packets because the drop command is situated in the header for the intermediary node. By dropping the packets at random at a sufficiently large frequency, the timing correlation is reduced (Levine et al., 2004).
- **Gamma buffering defense**. This technique is designed to remove timing correlation from the network stream through limited delays at the proxies. Standard packet-level batching effectively intertwines the stream's timing characteristics. Some packets are buffered before they are passed over from one mix to another. Let p be the number of incoming connections to a

node and *g* the value of gamma, then the node must buffer a total of *g* x *p* number of packets. A disadvantage of the method is that it could lead to additional delays.

Sybil Attacks

Anonymous communication system and all other distributed communication systems are vulnerable to security attacks called Sybil attacks. An attacker pretends to have multiple identities called Sybil identities or Sybil nodes (Douceur, 2002). Sybil attacks can tamper with routing protocols and prevent legitimate network nodes to obtain network resources (Balachandran & Sanyal, 2012). Several techniques for defense against Sybil attacks have been proposed. In Balachandran and Sanyal (2012) eight techniques – Trusted Certification, Resource Testing, Recurring Fees, Priviledge Attenuation, Economic Incentives, Location/Position Verification, Received Signal Strength Indicator (RSSI) based scheme, and Random Key Predistribution – are reviewed.

It can be proved that trusted certification is probably the only method, which provides complete defense against Sybil attacks (Doucour, 2002). However, this method requires the presence of a trusted Certification Authority (CA) for unambiguous validation of the correspondence between a network node and its associated identity. The problem of creating a trust relationship between two communicating parties is thus eliminated by the availability of a centralized CA.

Resource testing is possibly the most common approach to defense against Sybil attacks without services of a centralized CA. The basic principle is limitation of the number of accepted identities in a network. The proposed resource testing methods therefore only reduces the risk of Sybil attacks. For example, the proposed resource testing method SybilLimit limits the highest possible number of Sybil nodes in a social network to $O(g \cdot \log(n))$, where n is the number of network nodes and g is the number of trust relationships between a legitimate node and a Sybil node (Yu, Gibbons, Kaminsky, & Xiao, 2008).

A localization algorithm using a RSSI based scheme can achieve 100% detection of Sybil nodes with only a few false positives, but only in wireless sensor networks (Demirbas & Song, 2006). Sybil nodes in a wireless network can also be detected with a related method, Radio Device Fingerprinting, since each physical node in a wireless network has a unique radio fingerprint (Sieka, 2006).

The 2G Onion Router – TOR

The Second-Generation Onion Router, also known as TOR, is the most popular onion routing system that is currently available. As previously mentioned, onion routing is meant to help both users and hosts to hide their location from other parties. Authorities can spy on traffic from anonymization servers by watching all incoming and outgoing connections, for example, all communication coming in and going out from a VPN provider. Since communication might be encrypted, endpoints can be revealed, and may expose who is behind the communication. By allowing anyone to join and leave the TOR network at any time while also onion routing is used, it is impossible to spy on all nodes in the network. Anyone can access the TOR network by downloading the TOR client. TOR Bundle is a modified version of the open source Web browser Mozilla Firefox (TOR Project, 2013).

In Figure 4 we can see an example of a node connecting to the TOR network through the client program. First the client connects to a catalog server that gives the client the IP address to an entry node in the TOR network. Step 2 represents onion routing where a server decrypts one part of the packet sent and gets the IP address to the next onion router. Once the routing is completed unencrypted traffic can be relayed from the exit node to the targeted server. Figure 5 represents the same client resetting the connection to get another route through the TOR network, but the steps are the same.

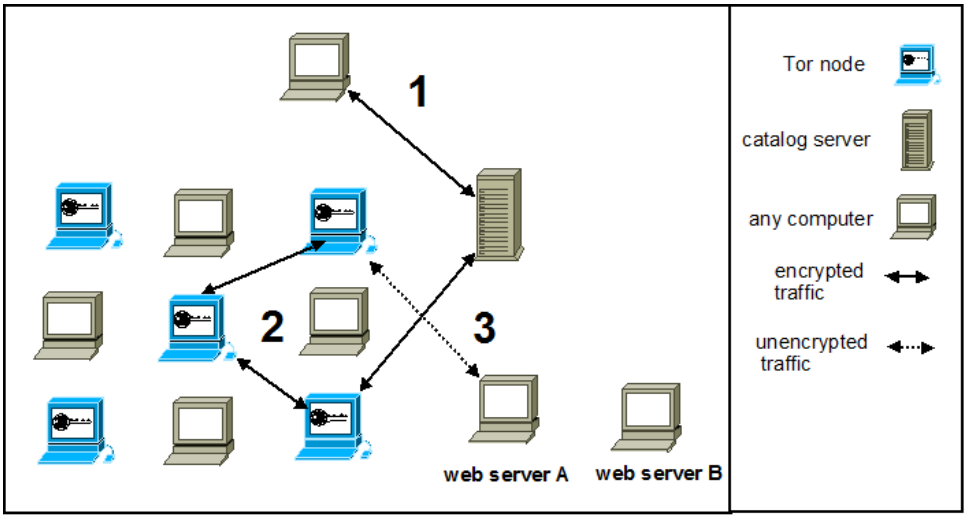

Figure 4: Example connection in TOR.

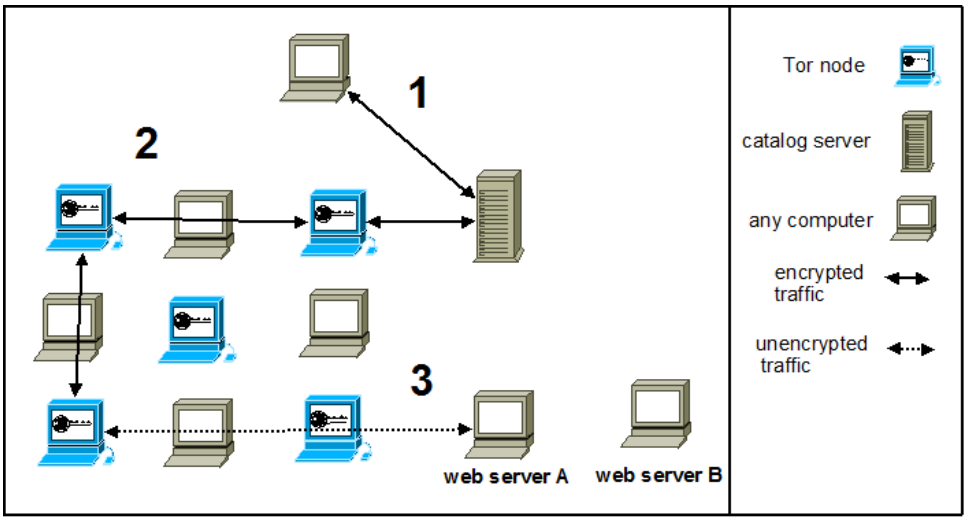

Figure 5: Resetting connection results in a new onion circuit.

Design

TOR uses onion routing to anonymize users and hosts. The anonymization works with TCP (Transmission Control Protocol) and can be used by any application that has SOCKS support. SOCKS stands for Socket Secure Protocol and is a part of the Internet protocol suite as a layer between the application layer and the transport layer. TOR addresses previous limitations by having new functionality including perfect forward secrecy, congestion control, directory servers, integrity checking, configurable exit policies and practical design for location-hidden services via rendezvous points (Dingledine, Mathewson, & Syverson, 2004).

Perfect forward security means that TOR uses an incremental path-building design. The initiator handles session keys for each hop in the routing. When keys are removed the following hops can no longer be decrypted even if the node that handled the traffic would be compromised. The TOR network allows clients at edges of the network to detect congestions and can therefore reduce data output until the congestion is resolved (Dingledine et al., 2004). Compared to earlier onion routing, the TOR network takes a simplified view towards sharing information. The network contains a number of trusted nodes that act as directory servers and provide signed directories that describe

known routers and states (Dingledine et al., 2004). TOR has added functionality for checking packet integrity so that no node can alter the data from one point to another in the network. Rendezvous points in the TOR network are negotiated by clients so that clients can connect to hidden servers; this provides forward security (Dingledine et al., 2004).

Forward security means that even if one of the parts in the onion routing is compromised it cannot be used to divulge information about the other communication in the network. To avoid traffic capturing TOR uses Diffie-Hellman key exchange between the onion proxy and each circuit router for the duration of a circuit's lifetime. Variable exit policies are allowed, i.e. each node has a policy describing the hosts and ports to which it will connect. Some features that are considered to be unnecessary in TOR such as mixing, padding and traffic shaping are not included (Wiangsripanawan, Susilo, & Safavi-Naini, 2007).

TOR allows users to setup hidden services inside the network. To get a hidden service running a user must have a TOR client installed and must install a web server locally on the same computer. Some configuration is needed to insure that the server does not give away any relevant information and stays anonymous. Once the service is configured the TOR client will configure a public key and a private key for the server so that the traffic to it can be encrypted (TOR Project, 2013).

Protocol

TOR is a tunneling protocol spanning over multiple nodes. A tunnel (circuit) is established incrementally hop by hop. Every connection is TLS encrypted and typically TCP port 443 is used. Traffic that needs to be tunneled and anonymized is generated by client nodes. Relay nodes route the traffic. The circuit traffic is basically sent in fixed packets (cells) and variable length cells are used for padding purposes. The general structure is shown in Figure 6. In the cell, the fixed cell length is 512 bytes. Circuit setup and teardown, protocol version negotiation, data relay, and peer authentication and authorization are handled by the commands. (Comaneci, 2013)

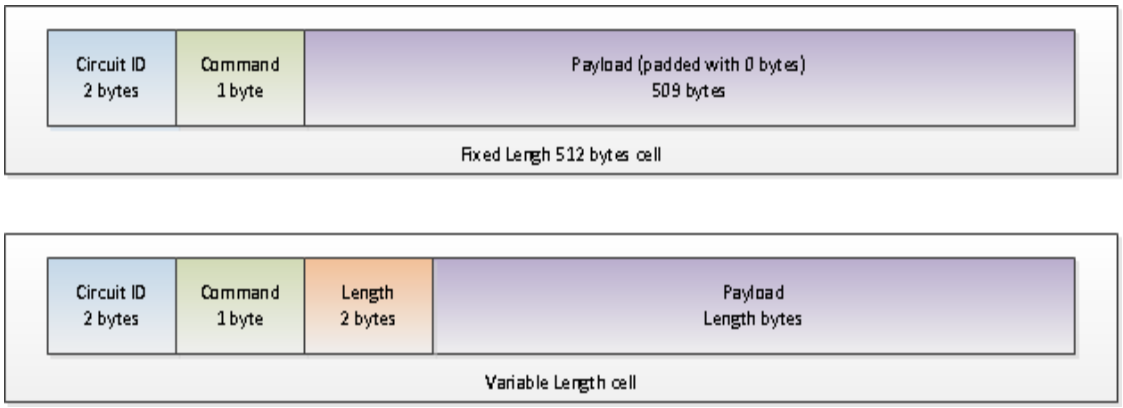

Figure 6: TOR cell structure (Comaneci, 2013).

In onion routing, one circuit was built for each TCP stream, but in TOR, each circuit can be shared by many streams. The old onion routing used a stream cipher without integrity checking. This vulnerability has been removed because TOR uses TLS on its links, i.e. external adversaries cannot modify data. Changes in network topology and node state are handled by directory servers (Dingledine et al., 2004).

Circumvention

TOR is a tunneling protocol spanning over multiple nodes. A tunnel (circuit) is established. A large number of users have chosen TOR not only for its anonymity properties but for its censor-

ship resistance properties. User access to Internet sites is therefore no longer affected by local censorship and firewall rules. The TOR design can be affected by blocking the directory authorities, by blocking the relay IP addresses in the directory or by filtering based on the network fingerprint of the TLS handshake (Dingledine & Mathewson, 2006).

TOR aims to provide three security properties:
1. The attacker can't learn or affect the destination.
2. No single network router can link you to the destination node.
3. The destination, or somebody watching it, can't know your location.

Here we present some other features present in the TOR design (Dingledine & Mathewson, 2006):

- TOR directory authorities automatically aggregate, test and publish signed summaries of the available routers.
- Clients use the default authorities if no others are specified.
- Given a set of routers TOR takes care of building paths through the network, not the user.
- The role of internal relay is separated from the role of exit relay.
- TOR is sustainable.
- Any single relay is prevented from linking users to their communication partners.

Relay based blocking resistance schemes generally have two main components: a relay component and a discovery component. The relay part establishes a connection and sends data. Discovery is the process of finding one or more usable relays. Many commercial methods are based on centrally-controlled shared proxies, independent personal proxies or open proxies.

Threat Model

In this section we go through threats against TOR. TOR has implemented a trusted node system where the entry node to the network has a high likelihood of being in a list of trusted nodes (Dingledine et al., 2004). This means that the passive predecessor attacks are prevented with a high likelihood. TOR is susceptible to intersection attacks as discussed earlier in the article.

Timing attacks is a real threat in TOR. There is currently no option of protection against timing attacks due to the latency increases induced by using a protective system as the one that Levine & al (2004) suggests. The general idea in TOR has been to protect against learning between which nodes there is communication, not to protect against confirmation if two nodes are communicating (TOR Project, 2013). The TOR network is protected against Sybil attacks by limiting the number of nodes that are allowed per IP address (TOR Project, 2013). However, this does not protect against having multiple nodes from different geographic locations. One way of protecting against having many malicious nodes in different locations is to have a trusted list of nodes for the entry nodes. Another protection against Sybil attacks is having a large user base, so that even if an attacker has many malicious nodes in the network, the chance of getting several malicious nodes in the same route is low.

Due to its architecture TOR is also susceptible to browser based and low resource routing attacks. In a low resource routing attack (Bauer, McCoy, Grunwald, Kohno & Sicker, 2007) a node in the TOR network misrepresents its network traffic capacity in order to increase latency in the network. To avoid this type of attacks the authors presented an algorithm that can be used to analyze the capacity of each node instead of having each node to decide its own capacity. A browser based attack is a combined attack of a man-in-the-middle attack and a Web browser's code execution attack (Abbott, Lai, Lieberman, & Price, 2007). To prevent browser based attacks users have to make sure that extensions are not allowed in the browser they are using. If a user uses the standard TOR bundle and runs the program from a virtual machine that is reset after each execution of the program, then the browser based attack should be avoided (TOR Project, 2013).

Since 2007, the TOR system consists of four node types: publicly listed relays, clients, bridges, and censored nodes. Publicly listed relays (onion routers) relay traffic between end users and the rest of the Internet. Clients connect to TOR through public relays and bridges are clients that act as unlisted relays for censored nodes (McLachlan & Hopper, 2009).

In TOR, clients download a list of servers from central directory authorities. This makes them an attractive target for attackers. Another problem is the use of a relatively small number of servers to create anonymous circuits for a huge number of users. The architecture also requires the users to maintain a global view of the system. A large number of servers make the system expensive (Mittal & Borisov, 2012).

Access Control

There are six different scenarios where a user with malicious intent can use the anonymity that TOR provides: 1) user spreading spam or malicious content such as viruses and malware, 2) spying on network communication, 3) planning of domestic or foreign terrorist attacks, 4) hosting of pornographic material of minors, 5) spreading of hate propaganda, 6) users taking advantage of anonymity when hacking into systems.

Here we present methods that could be incorporated into TOR in order to make it harder to use anonymity for malicious purposes. The presented techniques are: blacklisting, whitelisting, spam marking and Web filtering.

Blacklisting

Blacklisting is used to prevent known malicious nodes to communicate inside a network. There are different kinds of blacklisting, although generally all blacklisting systems have a database of nodes from which any communication is refused (Tiirmaa-Klaar, Gassen, Gerhards-Padilla, & Martini, 2013). Blacklisting could be used in TOR to prevent known assailants from connecting to the network. TOR network nodes have port 25 disabled, which is the standard port through which emails are sent, and use of this port for spamming through TOR is therefore not an issue (TOR Project, 2013). Known spammers and other known malicious users could still be prevented from accessing the network by implementing blacklisting.

Whitelisting

Whitelisting is considered to be the exact opposite to blacklisting. This means that only the users or nodes authorized in a database are allowed to connect to a network (Erickson, Casado, & McKeown, 2008). In general this would mean that in the TOR network new users would have a hard time connecting to the network if a whitelist determines who can use the network. However, if the whitelist would only be used to determine who can send email messages from the TOR network instead of determining who could access the network, then new users could still connect to the network.

Spam Marking

Spam marking is the procedure of analyzing email that is received and sent by users in a network to find and prevent spam. These techniques are also known as anti-spam (Anti-spam, 2013). The analysis can be text based, in which the content of the email will be analyzed. The analysis can also be based on behavior. In that case the system tries to find patterns in sending emails that will with a high likelihood be spam. A simple technique has been implemented to avoid spamming of email through the TOR network. Email is normally sent through port 25 on a computer and this port is by default closed on exit nodes. This means that no outgoing email can be sent through

normal nodes in the network, though it is still possible to connect to other services that allow sending of emails.

Web Filtering

There are several different versions of Web filtering. Some Web filtering is done based on IP addresses, some filtering is based on DNS (domain name system) names, and some filtering is content based (Chen & Wang, 2010). Content based filtering requires an active agent, like a program to be filtering each Web page a computer visits and as such would impose hardware constraints that most likely are not acceptable in a system like the TOR network. IP address based filtering wouldn't either be useful in the TOR network due to its anonymous architecture. However, a domain name system based on content filtering could be implemented since the TOR network uses a built-in DNS to connect to hidden services. A content filter based on DNS filtering could then reject communication to a DNS name that hosts unwanted content like pornographic material of minors or websites spreading hateful propaganda.

Conclusions

Much research is going on in the field of anonymous communication. Research on weaknesses in current systems as well as on protection against known and unknown attacks is important. Three different categories of attacks are found to be of importance: internal-external attacks, passive-active attacks and static-adaptive attacks.

A predecessor attack is a passive attack that is hard to detect. The predecessor attack can prevented by implementing onion routing with trusted entry nodes (Sylverson et al., 2000). Timing attacks are usually passive attacks but sometimes active attacks which try to find out who is communicating with whom by analyzing network traffic. To protect against timing attacks a system needs to implement rate cover traffic and random dropping (Levine et al., 2004). Intersection attacks are passive attacks which try to eavesdrop on traffic and find out the receiver of traffic. To defend against intersection attacks a system needs to implement some sort of buddies' architecture (Wolinsky et al., 2013).

The Sybil attack is a static and adaptive attack where one user controls many user identities in a network. It is possible to have protection against Sybil attacks by the use of a Certification Authority (Balachandran & Sanyal, 2012). However in practice the mostly used defense against Sybil attacks is to limit the amount of nodes the same user can have in a network linked to one IP address.

Anonymity on the Internet is provided by many different services and programs. The first thing to notice is that each service and/or program has a specific area in which it provides anonymity. No analyzed service/program prevents all possible attacks. For example TOR protects against predecessor attacks and limits the possibilities of a Sybil attack, but does not protect against intersection and timing attacks.

The available systems and/or programs can also have specific attacks that only apply to those systems. For example TOR is vulnerable to browser based attacks if the user doesn't follow the correct instructions as well as to Low-Resource attacks (Bauer et al., 2007) due to its current architecture.

Anonymous services and/or programs perform a vital task. Oppressed individuals can communicate without risking imprisonment and whistleblowers can send information to news stations without being intercepted. Ordinary people can also use anonymous services to secure connections to for example bank services or simply to make sure that their movement on the Internet is not recorded. It is even recommended for everyone to use some form of encryption when using

the unprotected or unreliable wireless Internet because RAPs (Rogue Access Points) can be set up to listen to all unencrypted traffic (Noor & Hassan, 2013).

Anonymous services also enable ill-intentioned individuals to misuse anonymity to perform malicious acts. While it is hard to totally stop misuse of anonymous services there are a number of techniques that can be used to make it harder for users with malicious intent to perform their actions. By implementing blacklisting in anonymous services and/or programs we can make it harder for known ill-intentioned individuals to connect to an anonymous network. DNS level filtering in anonymous networks could make it harder to host unwanted content from hidden services.

Full anonymity on the Internet cannot be guaranteed. IP addresses are not linked to people; they identify computers (machines). Many services connect IP addresses to identity or behavior. For instance, telephone companies may associate IP addresses directly to names, addresses and credit card numbers.

Distributed anonymizing services such as TOR and I2P may grant a higher degree of anonymity and security than centralized services where a central point discloses one's identity. Still, collaborating eavesdroppers, intersection and timing analysis may compromise the anonymity.

In this paper, anonymity has been discussed from a technical point of view. General issues connected with pros and cons of being anonymous on the Internet are of major importance.

References

Abbott, T. G., Lai, K. J., Lieberman, M. R., & Price, E. C. (2007). Browser-based attacks on Tor. PET'07 *Proceedings of the 7th International Conference on Privacy Enhancing Technologies*, pp. 184-199. Berlin, Heidelberg: Springer-Verlag.

Anonymous web surfing and online anonymity solutions anonymizer. (2013). Retrieved from http://www.anonymizer.com

Anti-spam Research Group (ASRG). (2013). Retrieved from http://asrg.sp.am

Balachandran, N., & Sanyal, S. (2012). A review of techniques to mitigate Sybil attacks. *International Journal of Advanced Networking and Applications, 4*(1), 1514-1518.

Bauer, K., McCoy, D., Grunwald, D., Kohno, T., & Sicker, D. (2007). Low-resource routing attacks against tor. *Proceedings of the 2007 ACM workshop on Privacy in electronic society,* pp. 11-20. ACM.

Bennett, K., & Grothoff, C. (2003). GAP – Practical anonymous networking. *Proceedings of Privacy Enhancing Technologies Workshop.* LNCS 2760, Berlin: Springer-Verlag.

Burke, J (2013). *NSA spied on Indian embassy and UN mission, Edward Snowden files reveal.* Retrieved from http://www.theguardian.com/world/2013/sep/25/nsa-surveillance-indian-embassy-un-mission

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms, *Communication of the ACM, 24*(2), 84–90.

Chaum, D. L. (1988). The dining cryptographers problem. *Journal of Cryptology 1*(1), 65-75.

Chen, T. M., & Wang, V. (2010). Web filtering and censoring. *Computer*, *43*(3), 94-97.

Clarke, I., Sandberg, O., Wiley, B., & Hong, T. W. (2001).Freenet: A distributed anonymous information storage and retrieval system. *Proceeding of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, New York: Springer-Verlag, 46-66.

Comaneci, D. (2013). *Protecting internet anonymity with TOR - The Onion Router Protocol*. Retrieved from http://blogs.ixiacom.com/ixia-blog/protecting-internet-anonymity-with-tor-the-onion-router-protocol

Corrigan-Gibbs, H., & Ford, B. (2010). Dissent: Accountable anonymous group messaging. *Proceedings of the 17th ACM Conference on Computer and Communications Security CCS'10*, pp. 340-350. ACM.

Daginawala, H., & Wright, M. (2008). Studying timing analysis on the internet with SubRosa. In *Privacy Enhancing Technologies*, LNCS 5134, Berlin: Springer-Verlag, 133-150.

Danezis, G., Dingledine, R., & Mathewson, N. (2003). Mixminion: Design of a type III anonymous remailer protocol. *Proceedings of the 2003 Symposium on Security and Privacy*, IEEE, 2-15.

Danezis, G., & Serjantov, A. (2005). Statistical disclosure or intersection attacks on anonymity systems. *Information Hiding*, LNCS 3200, Berlin: Springer-Verlag, 293-308.

Demirbas, M., & Song, Y. (2006). An RSSI-based scheme for Sybil attack detection in wireless sensor networks. *Proceedings of 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks WoWMoM*.

Dingledine, R. & Mathewson, N. (2006, November). *Design of a blocking-resistant anonymity system*. Technical Report 2006-11-001, The Tor Project. Retrieved from https://research.torproject.org/techreports/blocking-2006-11.pdf

Dingledine, R., Mathewson, N., & Syverson, P. (2004). *Tor: The second-generation onion router*. Naval Research Lab, Washington DC.

Douceur, J. R. (2002). The Sybil attack. *Proceeding of the 1st International Workshop on Peer-to-Peer Systems*, 251-260.

Egger, C., Schlumberger, J., Kruegel, C., & Vigna, G. (2013). Practical attacks against the I2P network. *Proceedings of the 16th International Symposium on Research in Attacks, Intrusions and Defenses*, St. Lucia.

Erickson, D., Casado, M., & McKeown, N. (2008). The effectiveness of whitelisting: A user-study, *Fifth Conference on Email and Anti-Span CEAS*. Microsoft Research, Silicon Valley, California, USA. Retrieved from http://www.ceas.cc/2008/papers/ceas2008-paper-20.pdf

Freedman, M. J., Sit, E. Cates, J., & Morris, R. (2002). Introducing Tarzan, a Peer-to-Peer Anonymizung Network Layer. *Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems IPTPS'02*, Cambridge, MA, USA. Retrieved from http://www.iptps.org/papers-2002/182.pdf

Garden Networks for Information Freedom. (2013). *About GTunnel*. Retrieved from http://gardennetworks.org/products

Gedik, B., & Liu, L. (2008). Protecting location privacy with personalized K-Anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing, 7*(1), 1-8.

Gercke, M. (2012). *Understanding cybercrime: Phenomena, challenges and legal response*. International Telecommunication Union ITU. Retrieved from http://www.itu.int/ITU-D/cyb/cybersecurity/docs/Cybercrime%20legislation%20EV6.pdf

Gianchandani, P. (2012). *Timing analysis attacks in anonymous systems*. Retrieved from http://resources.infosecinstitute.com/timing-analysis-attacks

GNU's Framework for Secure Peer-to-Peer Networking. (2013). Retrieved from https://gnunet.org

Goel, S., Robson, M., Polte, M., & Sirer, E. G. (2003). *Herbivore: A scalable and efficient protocol for anonymous communication.* Technical Report TR2003-1890. Ithaca, New York: Cornell University, Computing and Information Science.

Goldschlag, D. M., Reed, M. G., & Syverson, P. F. (1996). Hiding routing information, *Proceedings of Information Hiding: First International Workshop*, 137–150.

Gruteser, M. & Grunwald, D. (2003). Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. *Proceedings of the International Conference on Mobile Systems, Applications, and Services MobiSys*, New York: ACM, 163-168.

Gülcü, C., & Tsudik, G. (1996). Mixing email with Babel. *Proceedings of the Symposium on Network and Distributed System Security,* IEEE, 2-16.

How Anonymizers Work (2011). Retrieved from http://www.livinginternet.com/i/is_anon_work.htm

I2P Anonymous Network. (2013). Retrieved from http://www.i2p2.de

JAP Anonymity & Privacy (2011). Retrieved from http://anon.inf.tu-dresden.de/index_en.html

Joshi, P. (2009, March 21). Every move you make, Google will be watching you. *Business Standard.* Retrieved from http://www.business-standard.com/article/beyond-business/every-move-you-make-google-will-be-watching-you-109032100064_1.html

Kesdogan, D., Egner, J., & Büschkes, R. (1998). Stop-and-Go-MIXes providing probabilistic anonymity in an open system. In D. Aucsmith (Ed.), *Information hiding* (pp. 83-98). LNCS 1525, Berlin: Springer-Verlag.

Levine, B. N., Reiter, M. K., Wang, C., & Wright, M. K. (2004). Timing attacks in low-latency mix-based systems, In A. Juels (Ed.), *Financial cryptography* (pp. 251-265). LNCS 3110, Berlin: Springer-Verlag.

Levine, B. N., & Shields, C., (2002). Hordes – A multicast based protocol for anonymity. *Journal of Computer Security, 10*(3), 213-240.

Liang, Z.-W., Li, J., Li, C.-R., & Deng, J.-C. (2012). The survey of location privacy protection. *Proceedings of International Conference on Wavelet Active Media Technology and Information Processing ICWAMTIP*, 227 – 230.

Ling, Z., Fu, X., Jia, W., Yu, W., Xuan, D., & Luo, J. (2012 July). Novel packet size based covert channel attacks against Anonymizer. *IEEE Transactions on Computers*, 09.

Marques, R., & Zuquete, A. (2011). A social networking for anonymous communication systems: A survey. *Proceedings of the International Conference o Computational Aspects of Social Networks CASoN*, 249 – 254.

Mathewson, N., & Dingledine, R. (2004). Mixminion: Strong anonymity for financial cryptography. In A. Juels (Ed.), *Financial cryptography* (pp. 227-232), LCNS 3110, Berlin: Springer-Verlag.

McLahlan, J., & Hopper, N. (2009). On the risks of serving whenever you surf**.** Vulnerabilities in Tor's blocking resistance design. *Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society WPES'09*, ACM, 31-40.

Mislove, A., Oberoi, G., Post, A., Reis, C., Druschel, P., & Wallach, D. S. (2004). AP3: Cooperative, decentralized anonymous communication. *Proceedings of the ACM SIGOPS European Workshop*, ACM.

Mittal, P., & Borisov, N. (2012). Information leaks in structured peer-to-peer anonymous communication systems. *ACM Transactions on Information and System Security*, 15(1), 5:1-5:28

Mixmaster. (2012). Retrieved from http://mixmaster.sourceforge.net

MLSA. TOR (The Onion Router) (2005). Retrieved from http://webapps.lsa.umich.edu/lsait/admin/TOR%20Routing%20Infomation%20.pdf

Mute Simple, Anonymous File Sharing. (2013). Retrieved from http://mute-net.sourceforge.net

Nambiar, A., & Wright, M. (2006). Salsa: A structured approach to large-scale anonymity. *Proceedings of the 13th ACM Conference on Computer and Communications Security CCS'06*, New York, NY, USA: ACM, 17–26.

Noor, M. M., & Hassan, W. H. (2013). Current threats of wireless networks. *Proceedings of the Third International Conference on Digital Information Processing and Communications ICDIPC2013.* The Society of Digital Information and Wireless Communication, 704-713.

Palme, J., & Berglund, M. (2004). *Anonymity on the Internet.* Retrieved from http://people.dsv.su.se/~jpalme/society/anonymity.pdf

PLANETLAB An open platform for developing, deploying, and accessing planetary-scale services. (2007). Retrieved from http://www.planet-lab.org

Raymond, J.-F. (2000, July). Traffic analysis: Protocols, attacks, design issues, and open problems. In H. Federrath (Ed.), *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, LNCS 2009, Berlin: Springer-Verlag, 10-29.

Reed, M. G., Syverson, P. F., & Goldschlag, D. M. (1998). Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, *16*(4), 482–494.

Reiter, M. K., & Rubin, A. D. (1998) Crowds: Anonymity for web transactions. *ACM Transactions on Information System Security, 1*(1), 66–92.

Rennhard, M., & Plattner, B. (2002). Introducing MorphMix: Peer-to-peer based anonymous internet usage with collusion detection. *Proceedings of the Workshop on Privacy in the Electronic Society WPES*. Washington, DC, USA, 91–102.

Shmatikov, V., & Wang, M.-S. (2006). Timing analysis in low-latency mix networks: Attacks and defenses. *Proceedings of the 11th European Symposium on Research in Computer Security ESORICS*, Hamburg, Germany: Springer-Verlag, LNCS 4189, 18-33.

Sieka, B. (2006). Using radio device fingerprinting for the detection of impersonation and Sybil attacks in wireless networks. *Proceedings of the Third European Conference on Security and Privacy in Ad-Hoc and Sensor Networks*. Berlin, Heidelberg: Springer-Verlag, 179-192.

SmartHide Overview (2013). Retrieved from http://smarthide.com/overview

Syverson, P., Tsudik, G., Reed, M., & Landwehr, C. (2000).Towards an analysis of onion routing security. In H. Federrath (Ed.), *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, LNCS 2009, Berlin: Springer-Verlag, 96-114.

Tiirmaa-Klaar, H., Gassen, J., Gerhards-Padilla, E., & Martini, P. (2013). *Botnets,* SpringerBriefs in Cybersecurity, Berlin: Springer-Verlag.

Tor project. (2013). Retrieved from https://www.torproject.org

Wang, X., Chen, S., & Jajodia, S. (2007). Network flow watermarking attack on low-latency anonymous communication systems. *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, USA: IEEE Computer Society, 116-130.

Wiangsripanawan, R., Susilo, W. and Safavi-Naini, R. (2007). Design principles for low latency anonymous network systems secure against timing attacks. *Proceedings of the Fifth Australian Symposium on ACSW Frontiers – Volume 68*, Australian Computer Society, 183-191.

Wolinsky, D. I., Syta, E., & Ford, B. (2013). Hang with your buddies to resist intersection attacks. *Proceedings of the ACM Conference on Computer and Communications Security CCS*, ACM.

Wright, M., Adler, M., Levine, B. N., & Shields, C. (2004). The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security (TISSEC)*, *4*, 489–522.

Yu, H., Gibbons, P. B., Kaminsky, M., & Xiao, F. (2008). Sybillimit: A near-optimal social network defense against Sybil attacks. *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, Washington, DC, USA: IEEE Computer Society, 3-17.

Your Freedom Opens Your Door to the World. (2013). Retrieved November 22, 2013 from https://www.your-freedom.net

Zhu, B., Wan, Z., Kankanhalli, M. S., Bao, F., & Deng, R. H. (2004). Anonymous secure routing in mobile ad-hoc networks. *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks LCN'04*, IEEE, 102-108

Biographies

Kaj J. Grahn received his doctoral degree at Helsinki University of Technology and is presently senior lecturer in information technology at Arcada University of Applied Sciences, Helsinki, Finland. His current research interests include security of wireless and mobile networks.

Thomas Forss is a Ph.D. student at the department of Information Systems at Åbo Akademi University. He has also completed a Master's degree in Software Engineering at Åbo Akademi. Currently he works as a lecturer and researcher at Arcada University of Applied Sciences. His current research interest is in the area of analytics.

Göran Pulkkis received in 1983 his doctoral degree at Helsinki University of Technology and is presently project researcher in computer science and engineering at Arcada University of Applied Sciences, Helsinki, Finland. His current research interests include network security and applied cryptography.