# Anonymity in Networks I - Basics

Janosch Maier
Supervisor: Heiko Niedermayer
Proseminar Network Hacking und Abwehr WS2011
Chair for Network Architectures and Services
Fakultät für Informatik, Technische Universität München
Email: maierj@in.tum.de

## ABSTRACT

This paper introduces important terms related to anonymity, describes basic attacks on people's identity in networks and proposes defense mechanisms.

Different use cases of networks need different approaches in anonymizing. Due to the need of real-time answers in some applications they have different requirements. Mixing and onion routing are concepts that try to create anonymity in networks. Various anonymizers use those techniques to provide anonymity for the purpose of – amongst others – sending E-mails, web browsing and chatting. Some of those porgrams are briefly depicted and compared, addressing problems and possible security breaches.

## Keywords

Anonymity, Pseudonymity, Identity, Internet, Network, Privacy, Security

## 1. INTRODUCTION

Sending and receiving data via a secure connection does not mean that you are acting anonymously [29]. If you access a website like `https://wikileaks.org` a SSL/TLS-encrypted connection is used. In this case your network administrator cannot see the content you see [21]. The three packages in Listing 1 were monitored by the freely available tool wireshark [1]. The last package shows the used encryption.

```
Protocol  Info
DNS       Standard query A wikileaks.org
DNS       [...] query response A 88.80.2.31
TLSv1     Server Hello
```

**Listing 1: DNS request and encrypted HTTP request**

Nevertheless looking at those packages (Listing 1), it is visible that you have made a DNS-request on wikileaks.org and later accessed it.

Given: Alice is a member of a law enforcement agency. She has identified a website used for exchanging information by a group of computer criminals. She accesses it repeatedly from her office computer to gather information (Figure 1). If Eve – a member of the criminal group and responsible for the website – can trace several connections back to Alice, he will just shut down the service. Therefore Alice's chances of catching those criminals decreases a lot.


**Figure 1: Investigators vs. Computer Criminals**

Also for people who have no special reason to stay anonymous, there is technically no need to be identified by a communication partner. Somebody browsing on public websites or sending emails has no necessity to reveal his identity.

Chapter 2 describes some important terms needed to understand anonymity in networks. Chapter 3 shows attacks on people's data. The next chapter 4 focuses on defense mechanisms to the stated attacks. It presents the mechanisms onion routing and mixing and compares those techniques. Chapter 5 takes a look at the problems that cannot be solved using these mechanisms. In chapter 6 related works are presented and briefly evaluated. The last chapter 7 concludes this paper and shows which anonymizing method is useful in which case.

## 2. DEFINITIONS

### 2.1 Networks

A node is a person or entity that is part of a network. A sender is a node sending any kind of data – called message – over a network. A receiver is a node receiving a message over a network. A hop is a node that a message passes while being transmitted from its sender to the receiver.

### 2.2 Anonymity and Pseudonymity

Anonymizing data [12] means changing personal data in a way that makes it either impossible or causes disproportionate effort connecting the data to one certain person. A dataset which does not contain any personal information to identify its originator is anonymized. In contrast, pseudonymity means replacing the name or other identifiers of data with some other identity – a pseudonym. Thus making identifying the person complicated or impossible, for everybody but a trusted authority. The trusted authority is the institution that knows both, the pseudonym and the real identity. Speaking of the internet, everybody with a private internet connection has his pseudonym – the IP address. The internet provider is the trusted authority, which

[1] `http://www.wireshark.org/`

can combine IP address and personal data to reveal somebody's identity. Knowing somebody's pseudonym – e.g. the IP address or a forum nickname – an attacker is able to create a profile of the person hidden by the pseudonym. Even though he does not know the real identity, the person using this pseudonym will be the same for a certain period of time.

## 2.3 Extends of Anonymity

Anonymity in networks can have different extends: Sender anonymity means that the sender of a message can not be traced back (Figure 2). Receiver anonymity means that it is not possible to find out which node received a certain message (Figure 3). Unlinkability means that it is not possible for the attacker to determine which nodes are communicating with each other (Figure 4). With more than one anonymously sent message this means the attacker is unable to determine whether the messages are always sent and received by the same node [33]. To create any kind of anonymity a set of similar nodes is needed. Those nodes might or might not communicate. Such a set is called anonymity set.


**Figure 2: Sender Anonymity**


**Figure 3: Receiver Anonymity**


**Figure 4: Unlinkability**

## 2.4 Attacks on Personal Data

Attacker stands for a person that has an interest in personal data of someone. This personal data may be the type of communication, communication patterns, participants of a communication or the single fact that there is communication. An attacker might also try to manipulate the data. The attacker might be the receiver of a message, for example the website somebody is accessing or somebody in the sender's local network, his network administrator. A person somewhere else in the network [33] like a criminal prosecutor or a hacker is also possible.

# 3. BASIC ATTACKS

A user accessing a website leaves traces. The person controlling the website can easily obtain the IP address of the people visiting his site. In PHP, a command as simple as in Listing 2 will store your public IP address in a variable controlled by the PHP script [7].

```
$ip = $_SERVER['REMOTE_ADDR'];
```

**Listing 2: Sender IP address accessed using PHP**

The apache webserver is used to supply most of the websites that are available in the internet [13]. Showing the IP address of each page access in a log file can be configured easily [5].

Therefore if no anonymizing measures are taken it is easy for the receiver of a message to find out who he is communicating with. Taking the example of the computer criminals (Chapter 1) they might recognize an overproportional number of pageviews originating from the same – Alice's – node. Tracing back the IP address [16] to criminal investigators they could shut down their node before evidence is saved.

The example from Chapter 1 shows the power of a network administrator. Controlling the node connecting a local network with the internet traffic monitoring can be easily achieved using tools like tcpdump [2] or wireshark. Anonymity providing measures should disallow this procedure.

For any attacker gaining (root) access to the local network of the sender or the receiver, the same attack schemes are possible.

# 4. CREATING ANONYMITY

Given the global scope of such attacks anonymity usually means both sender and receiver anonymity. In the case of a random message in a network determining the sender or the receiver should be impossible. As the receiving node knows it is the receiver, receiver anonymity is not possible there. Sender anonymity is even more essential here. Looking at all the messages transferred in the network, unlinkability between sender and receiver is needed.

Some actions in a network might be delayed by an anonymizing system. For email a delay is not crucial. Other actions like web browsing need instant responses. For those actions real time anonymizing is important.

A service can be offered anonymously, therefore beeing accessible by a public address without the need to reveal information about the service. The needed approach will not be explained in this paper. Nevertheless this is possible by using onion routing (Chapter 4.2) [9] [1].

## 4.1 Mixing

### *4.1.1 Characterisation of Mixing*

Mixing is a system introduced for anonymous message delivery, that is highly suitable for mail systems [15]. Mails sent through this system are not sent directly to the receiver, but via a third node, a so called mix.

Each node in this mail system network – sender, receiver and mix – need to have an encryption key pair. Each key pair consists of a public key that is available for everybody and a private key that only the owning node knows. A message

[2] `http://www.tcpdump.org/`

encrypted with a public key can only be decrypted with the corresponding private key.

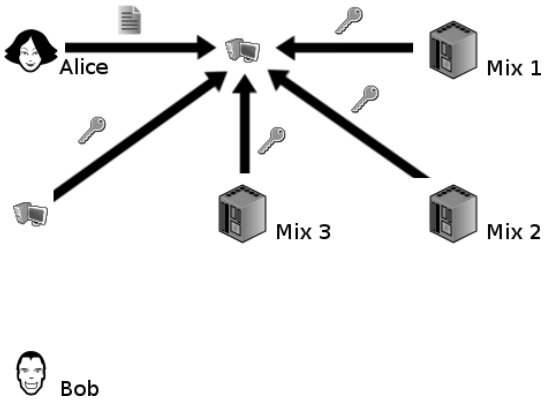


**Figure 5: Public keys are used to encrypt Alice's message multiple times**

Alice wants to send a mail to Bob. Alice encrypts her message plus a random string for cryptographic security with Bob's public key. The encrypted message and another random string are then encrypted with the public key of a mix (Figure 5). The message is delivered to the mix which decrypts the received message. It dumps the random string and then forwards the message to Bob. Bob is able to decrypt the received message and get the content of the mail. This guarantees that a certain message looks different before and after the mix. A linkage of an incoming and an outgoing message of one mix is not possible looking at its appearance.

The mix does not forward messages immediately. It waits until it has received and stored a configured amount of messages. It mixes those messages and delivers them as a batch. Therefore the time is no promising characteristic to connect an incoming with an outgoing message.

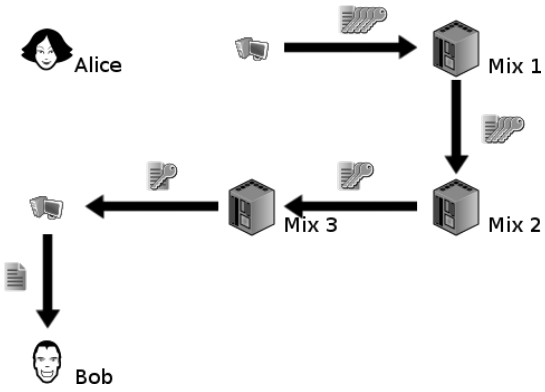


**Figure 6: Alice's message is sent to Bob through a mix cascade**

As – following this guide – the mix knows both the sender and the receiver of the message anonymity is not yet accomplished. A message does not have to be only sent via one mix. If a message takes a path of more than one mix, this is called a mix cascade. Each mix node is only able to see the sending and the receiving node for this message hop (Figure 6). The first mix only knows the real sender, the last mix only the real receiver. All mixes in between only know other mixes in the cascade. The goals stated at the beginning of chapter 4 are achieved.

To clarify this method one can think of encryption as putting a letter in an envelope that can only be opened by its addressee. Alice wants to send a letter to Bob. She takes the letter and puts it in an envelope addressed to Bob. She then puts the envelope in another envelope addressed to a post station (which acts as a mix). This envelope is put in another envelope addressed to another post station. Depending on the level of anonymity Alice needs, this can be done several times. All the post stations in a row are the mix cascade. Alice then delivers her package of envelopes to the first post station. There the outer envelope is removed. When the post station has received enough letters it forwards all letters to all their receivers at the same time. The next post station receiving Alice's letter does the same. This is repeated until Bob gets the letter, removes the last envelope and reads the message Alice has sent.

This works fine for sending messages in one direction. If Bob wants to answer Alice directly, he cannot do this, as he does not know her identity. So Alice needs to create a return address in a way that Bob cannot learn Alice's identity from it. Alice encrypts her address with the public key of the first node in the mix cascade. The first node adds its address and encrypts the package with the public key of the next node. This can be done for each node as it knows the identity of the next node in the cascade [15]. The return address is then forwarded in the same way as the actual message is. The succeeding node on the way back is, when an answer message is sent, the predecessor of the way forward in the mix cascade. Each mix can then decrypt the address for the next – and only the next – node.

### *4.1.2 Implementations of Mixing*

Remailer systems [29] – which can be divided into four types – use mixing to provide anonymity for sending E-mails. A type 0 remailer is an pseudonymous remailer. Type I is an anonymous remailer without the possibility to reply to an E-mail. Type II remailers allow to send answer messages. For those type of remailer systems a special mail program is needed. Type III remailer systems [17] introduce dummy traffic [25] to reduce the possible attacks.

Mixminion [3] is a type III remailer program that uses mixing to send mails via different mix nodes. It uses different directory servers to create a listing of all available mix servers. The Mixminion client chooses a path through the network according to the available servers on this list. Answer messages are implemented using so called Single Use Reply Blocks (SURBs) [25]. Messages between the nodes in the network are transported via an TLS encrypted connection [19].

JAP [4] or its premium version JonDo [5] use mixing [6] to create anonymity for real-time web browsing. To make mixing

[3] `http://mixminion.net/`
[4] `http://anon.inf.tu-dresden.de/`
[5] `http://www.anonym-surfen.de/`

work properly a certain amount of messages is needed. This interferes with the real-time experience that is needed for web browsing, and will be discussed in chapter 4.3.

## 4.2 Onion Routing

### 4.2.1 *Characterisation of Onion Routing*

Onion routing is also based on the idea of routing messages over different servers. It varies from mixing in a way that allows communication in both directions for low latency applications [25]. The onion routing network consists of onion routers that can pass messages using permanent socket connections between each other. Each router works as a mix that is nearly working in real-time [34].

Application data is given to an onion proxy which works as an entry point to the onion routing network. It defines a way through the network of onion routers. The data is encrypted multiple times in a way that resembles onion skins. Each layer of this onion defines one node on the path of the message. It contains information about the next hop and key material for encryption and decryption of data that is sent using this connection. The first node on the way can remove the outer layer of encryption passing the inner onion to the next node which can then encrypt the next layer. This is done until the last onion router can decrypt the innermost layer of encryption [35] and pass the plaintext message to its receiver. Data that is sent back from the receiver of the message is ecrypted in reverse order. The same connection that is used to pass messages to the receiver is used to return messages to the original sender.

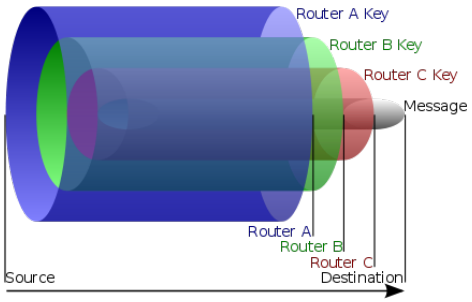


**Figure 7: Different Layers of an onion [32]**

Messages in the onion routing network appear differently at each node, as each router removes one layer of encryption. So data cannot be traced back by its appearance. Even compromised nodes working together cannot learn anything from these onions.

### 4.2.2 *Implementation of Onion Routing*

Tor (The Onion Router) [6] is a software project that uses onion routing to anonymize the identity of its users [11]. The Tor client is used only as onion proxy by default. It can be configured to act as an onion router in the onion routing network.

I2P [7] is a network that uses onion routing to create anonymity for its users [3]. It implements end-to-end encryption within the network. It works for web browsing, as well as sending emails and downloading torrents [2].

## 4.3 Comparison

Mixing and onion routing are different approaches to network anonymity, both being highly suitable for different kinds of applications.

By design mixing is convenient for mailing. This resulted in the development of different remailer systems using this technique for anonymizing purposes. Mixminion has – by theorie – the most sophisticated anonymizing protocol [25]. Dummy messages [17] created by the different mixes complicate attacks. Especially, when there is only low usage of the Mixminion network this covers the real messages received and send by a mix. By this day Mixminion still has a release version number that is below 1.0. Its readme file includes the following warning:

> "WARNING! Do NOT use this release if you require strong anonymity. It has known deficiencies, including some that make it possible for an adversary to trace your message through the system." [30]

The lack of testing and security assessment on Mixminion is most problematic here.

Mixmaster [8] is the main implementation of the type II remailer protocol of the same name. Its concept is not as advanced as the Mixminion protocol but at least it is widely tested. The only known successful attack model includes an attacker who can eavesdrop all remailer nodes and even intervene in the network [25].

JAP makes mixing usable for real-time applications. To keep the latency of the network low JAP has used the following approach: Packages from different JAP instances arriving at the first Mix node in a cascade are multiplexed and sent in a serialized way to the second node in the cascade [6]. The more messages are stored and mixed before forwarded to the next node, the more unlikely it becomes for an attacker to link incoming and outgoing messages. This delay results in a worse user experience of the service. This lets user switch their cascade or the whole service, decreasing the anonymity set [24].

Onion routing – and therefore Tor – is designed to provide anonymity in low latency networks. Tor does not introduce any kind of dummy messages [20] as Mixminion does to complicate attacks from an attacker with access to all nodes in the network. With the size of the network [10] – approximately 2500 running relay nodes by December 2011 – an attack on all nodes is very complicated. An attacker who can eavesdrop all the nodes, has most likely also the ability to get physical access to your computer. A trojan or a keylogger would rather be chosen in this scenario.

[6] `https://www.torproject.org/`

[7] `http://www.i2p2.de/`

[8] `http://mixmaster.sourceforge.net`

# 5. PROBLEMS

Though basic attacks that are described in chapter 3 can be avoided using those techniques more sophisticated attacks are succeeding. Those vary from method to method that is used for anonymization.

Intersection attacks [14] provide an attack scenario for which Tor does not aim to have a solution [20]. An attacker needs to watch all the messages all users send to a network as well as all messages that leave the network. He looks at a certain message and all users that have been active by the time the message has been sent to the network. Linking different messages to one session the attacker can reduce the anonymity set for this session significantly.

If the availability of one node in an anonymity network decreases users tend to use better routes through the network. An attacker could place some fast and highly available nodes in the network and sabotage other nodes to gain access to the information of a certain user. This might be possible for Mixminion [25]. How the anonymity of long-running services in the I2P network can be compromised by this attack has been shown already [22]. A simple DoS attack is used to replace mixes with ones own servers in order to determine the identity of a certain node in the network.

The static cascades of JAP make it possible for law enforcement to track the traffic of a certain user with jurisdictional order. Each provider of a mix in the used cascade has to get a warrant separately. This has been done a few times since the start of the network [4].

If no end-to-end encryption is used an attacker controlling a Tor exit node will see the plaintext of the messages sent. The Tor FAQ contains the following notice:

> "the guy running the exit node can read the bytes that come in and out there." [8]

By recording the traffic as in chapter 1 the attacker is able to get session cookies, username and password combinations and everything else that is included in the user's http request. This was used by the Swedish computer security researcher Dan Egerstad who posted [23] the login names and passwords of 100 E-mail accounts including those from embassies.

Main problems when using those methods to create anonymity is the user's unawareness or misunderstanding of those techniques.

Cookies that are set to identify a user are still in use if somebody is using anonymizing methods. Those cookies may allow the operator of a service to create profiles of a user, though he does not know the name or IP address. If the cookie has been set before any anonymizing technique has been used, a more detailed profile is possible. Connecting the formerly saved IP address and the recognized activities by the cookie, the effect of anonymization nullified.

Usernames that are used on more than one specific service are able to reveal a user's identity. Given: somebody uses one of the described anonymizing protocols. He logs into a service with his username. The same username is used on a different page or a social network, that has personal information about this user. Thus identifying the user is not difficult.

# 6. RELATED WORK

Various security analyses on different anonymity networks have been performed. Most papers focus on one highly theoretical attack on one special network. Solutions to those attacks are also provided there. But an overview over Anonymizing methods is mostly missing. These papers require a deep understanding of the techniques underlining the anonymity providing services. This work gives an introduction to these concepts.

The P2Priv [9] network introduces a way of anonymous Peer-2-Peer networks. It uses the anonymizing techniques described in chapter 4 to create cloning cascades for the purpose of anonymization. The initiator of the cloning cascade later communicates directly with the intended destination [28]. [27] suggests a parallelistic approach to further anonymization, called NetPriv. NetPriv also works in networks that have no large distribution of contents, which is one of the conditions for P2Priv to work properly.

[18] gives an introduction to traffic analysis. It describes its roots in military communication and attacks on every day use technologies of the modern internet. The paper states different possible counter measures but without explaining how and why those methods are working.

In [26] the term unlinkability is put in context of election systems. It shows how an electronic election system can be evaluated in terms of unlinkability of the single voter to his vote and verifiability of the election result.

[31] analyses tor hidden services. The difference in time of stated nominal and real clock frequency can be used as a fingerprint for one computer. This so called clock skew changes by temperature. Putting heavy load on a service increases the server's temperature resulting in clock skew changes. This can bea measured by an attacker. Using this technique the server providing a hidden servie can be deanonymized.

# 7. CONCLUSION

Without any measures providing anonymity an attacker can easily identify a user in a network. The concepts mixing and onion routing provide anonymity against basic attack schemes. Currently developed implementations of these techniques include Mixmaster, Mixminion, JAP and Tor. Those can be used to provide anonymity for different uses of networks. Remailers like Mixmaster and Mixminion based on the idea of mixing are suitable for an anonymous use of E-mails. JAP shows that mixing can also be used for near real-time applications. The main current implementation of onion routing – Tor – is designed for real-time applications and applicable for web browsing and chatting.

[9] `http://p2priv.org/`

As methods for anonymity arise more sophisticated attacks need to be run in order to obtain someone's personal data. The effort for these attacks is much higher than for basic attacks that are possible if no anonymizer is used. But a 100% security cannot be provided. When anonymity is needed the highest risk is the misunderstanding of anonymizing services. Mistakes a user can make can reveal his identity easily.

# 8. REFERENCES


[1] eepSite - I2P Anonymous and Secure Peer to Peer communication (P2P) Setup, Resource and Search Portal. http://www.eepsite.com/. [Online; accessed 25-January-2012].

[2] FAQ - I2P. http://www.i2p2.de/faq.html. [Online; accessed 26-January-2012].

[3] I2P Compared to Tor and Freenet - I2P. http://www.i2p2.de/how_networkcomparisons. [Online; accessed 26-January-2012].

[4] JonDonym law enforcement. http://anonymous-proxy-servers.net/en/law_enforcement.html. [Online; accessed 17-December-2011].

[5] Log Files - Apache HTTP Server. http://httpd.apache.org/docs/current/logs.html. [Online; accessed 12-December-2011].

[6] Mixes for Privacy and Anonymity in the Internet: Main Page. http://anon.inf.tu-dresden.de/develop/doc/mix_short/. [Online; accessed 12-December-2011].

[7] PHP: $_SERVER - Manual. http://de2.php.net/manual/de/reserved.variables.server.php. [Online; accessed 23-November-2011].

[8] Tor FAQ. https://trac.torproject.org/projects/tor/wiki/doc/TorFAQ. [Online; accessed 17-December-2011].

[9] Tor: Hidden Service Protocol. http://www.eepsite.com/. [Online; accessed 25-January-2012].

[10] Tor Metrics Portal: Network. http://metrics.torproject.org/network.html. [Online; accessed 17-December-2011].

[11] Tor Project: Overview. https://www.torproject.org/about/overview.html.en. [Online; accessed 30-January-2012].

[12] Bundesdatenschutzgesetz §3. http://www.gesetze-im-internet.de/bdsg_1990/__3.html, Dec. 1990. [Online; accessed 12-December-2011].

[13] November 2011 Web Server Survey | Netcraft. http://news.netcraft.com/archives/2011/11/07/november-2011-web-server-survey.html, Nov. 2011. [Online; accessed 29-November-11].

[14] O. Berthold and H. Langos. Dummy traffic against long term intersection attacks. In *PET'02 Proceedings of the 2nd international conference on Privacy enhancing technologies*, pages 110–128, Berlin, Heidelberg, Apr. 2002. Springer.

[15] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, Feb. 1981.

[16] L. Daigle. WHOIS Protocol Specification. http://tools.ietf.org/html/rfc3912, Sept. 2004. [Online; accessed 11-December-2011].

[17] G. Danezis. *Better anonymous communications*. Dissertation, University of Cambridge, Jan. 2004.

[18] G. Danezis and R. Clayton. Introducing traffic analysis. In A. Acquisti, S. Gritzalis, C. Lambrinoudakis, and S. D. C. di Vimercatio, editors, *Digital Privacy: Theory, Technologies, and Practices*, pages 95–116. Auerbach Publications, 2007.

[19] G. Danezis, R. Dingledine, and N. Mathewson. Type III (Mixminion) Mix Protocol Specification. http://mixminion.net/minion-spec.txt, Apr. 2007. [Online; accessed 14-December-2011].

[20] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium-Volume 13*, pages 303–320. USENIX Association, 2004.

[21] A. Freier, P. Kocher, and P. Karlton. The SSL Protocol Version 3.0. http://tools.ietf.org/html/draft-ietf-tls-ssl-version3-00, Nov. 1996. [Online; accessed 11-December-2011].

[22] M. Herrmann and C. Grothoff. Privacy-Implications of Performance-Based Peer Selection by Onion-Routers: A Real-World Case Study Using I2P. In S. Fischer-Hübner and N. Hopper, editors, *Privacy Enhancing Technologies*, volume 6794 of *Lecture Notes in Computer Science*, pages 155–174, Berlin, Heidelberg, 2011. Springer.

[23] M. Huber and M. Mulazzani. Tor HTTP usage and information leakage. In B. D. Decker and I. Schaumüller-Bichl, editors, *Communications and Multimedia, 11th IFIP TC 6/TC 11 International Conference, CMS 2010, Linz, Austria, May 31 – June 2, 2010. Proceedings*, volume 6109 of *Lecture Notes in Computer Science*, pages 245–255, Berlin, Heidelberg, 2010. Springer.

[24] S. Köpsell. Low Latency Anonymous Communication–How Long Are Users Willing to Wait? In G. Müller, editor, *Emerging Trends in Information and Communication Security, International Conference, ETRICS 2006, Freiburg, Germany, June 6-9, 2006. Proceedings*, pages 221–237, Berlin, Heidelberg, 2006. Springer.

[25] J. Kubieziel. *Anonym im Netz. Wie Sie sich und Ihre Daten schützen [Broschiert]*. Open Source Press; Auflage: 2. Auflage., München, 2010.

[26] L. Langer, H. Jonker, and W. Pieters. Anonymity and Verifiability in Voting: Understanding (Un)Linkability. In *Information and Communications Security*, pages 296–310, Berlin, Heidelberg, 2010. Springer.

[27] I. Margasiński. A Parallelism Based Approach to Network Anonymization. In A. Jøsang, T. Maseng, and S. J. Knapskog, editors, *Identity and Privacy in the Internet Age*, volume 5838 of *Lecture Notes in Computer Science*, pages 28–43, Berlin, Heidelberg, 2009. Springer.

[28] I. Margasinski and M. Pioro. A Concept of an Anonymous Direct P2P Distribution Overlay System. In L. O'Conner, editor, *22nd International Conference on Advanced Information Networking and Applications, 2008. AINA 2008.*, pages 590–597, Okinawa, 2008. IEEE Computer Society.

[29] D. M. Martin Jr. *Local Anonymity In The Internet*. Dissertation, Boston University Grade School of Arts and Sciences, 1999.

[30] N. Mathewson. README,v 1.83. `http://mixminion.net/dist/0.0.8alpha3/README-0.0.8alpha3`, Sept. 2007. [Online; accessed 17-December-2011].

[31] S. J. Murdoch. Hot or not: Revealing Hidden Services by their Clock Skew. In *Proceedings of the 13th ACM conference on Computer and communications security - CCS*, page 27, New York, USA, Oct. 2006. ACM Press.

[32] H. Neal. Onion diagram.svg. `http://de.wikipedia.org/w/index.php?title=Datei:Onion_diagram.svg&filetimestamp=20080815233751`, Mar. 2008. [Online; accessed 15-December-2011].

[33] A. Pfitzmann. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. `http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf`, Aug. 2010. [Online; accessed 09-November-11].

[34] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Onion routing network for securely moving data through communication networks. `http://patft1.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=/netahtml/PTO/search-bool.html&r=1&f=G&l=50&co1=AND&d=PTXT&s1=6266704.PN.&OS=PN/6266704&RS=PN/6266704`, May 1998. [Online; accessed 12-December-2011].

[35] P. Syverson, D. Goldschlag, and M. Reed. Anonymous connections and onion routing. In *1997 IEEE Symposium on Security and Privacy, May 4-7, 1997, Oakland, CA, USA*, IEEE Symposium on Security and Privacy, pages 44–54, Oakland, 1997. IEEE Computer Society.