Anonymous Communication

https://iuk.one/1033-1013

Clemens H. **Cap**
ORCID: 0000-0003-3958-6136

Department of Computer Science
University of **Rostock**
Rostock,Germany
clemens.cap@uni-rostock.de

Version 2

Overview

1. What is Anonymity?
2. Superposed Sending
3. Mix Networks
4. Remailers
5. Onion Routing
6. Further Remarks

1. What is Anonymity?

Understanding the concept and the necessity.

What is Privacy?

Possible Answers: 4 Doctrines of Privacy

Privacy in Private (Warren & Brandeis)

- Concept of privacy as "right to be left alone".
- Legal concept which as developed when photography was invented.

Privacy in Public (Volkszählungsurteil)

- Every person has the right to determine who has access to her personal data.

Interpersonal Privacy (Trading)

- Personal data can be traded for benefits (eg: Facebook: Free social network).

Zero Privacy (Post Privacy Society)

- "There is no privacy – get over it" (Scott McNealy)

Many Variants of Anonymity and Privacy

Many variants:

- Anonymous communication (this unit).
- Zero Knowledge Protocols.
- Secret Splitting and Secret Sharing.
- Multi Party Computation.
- Private Information Retrieval.
- Homomorphic Encryption.

What is Anonymity?

Answer 1: Not knowing an identity.

- Same problem as with "absolute security".
- Allows no quantification.
- Does not properly address notion of "identity".

Answer 2: Unlinkability

- I cannot link a communication act to context information.
- Examples: IP/MAC address, name, pseudonym, year of writing, used protocol.
- Solves the "identity" problem via "linkage".
- Still does not allow a quantification.

Answer 3: Size of anonymity set

- User is one out of a set with n elements.
- Example 1: Year of writing.
- Example 2: IP address of writer.
- Allows quantification by the probability with which information can be linked.

Use Cases for Anonymity

Abstract Use Cases

- Separating the message from the messenger.
- Anti censorship.
- No tracking.
- Escaping unwanted communication (spam).

Concrete Use Cases

- We are a ... dissident in ...
- We want to read ... material which is prohibited in ...
- We want to write ... material which is prohibited in ...
- We want to buy a product and not pay the highest price.
- We ... umm ... have something we want to hide.

Ethical Aspects of Anonymity

Pro: Philosophic position of enlightenment ("Aufklärung")

- Rational debate needs opportunity to state positions without detriment for messenger.
- Restrictions to open, anonymous communication damage democracy.

> Voltaire: "I might disagree with your opinion but I will fight that you can voice it freely."

Contra: Anonymous communication may be used to cover illegal activity.

- Use for distributing copyrighted, banned or illegal contents.
- Threats, blackmailing

Infrastructure Design Argument

- Building IT infrastructure that it strengthens human rights or promotes surveillance.

Technological Neutrality Argument

- Technology should not prejudice social and legal decisions.

Scenarios of Anonymity Quantification

Criminal court:

- "Beyond reasonable doubt"
- "In dubio pro reo"

Scenario 1:

- The probability of Alice being the sender (and thus guilty) is less than 50%.
- The probability of Alice being innocent is higher than of Alice being guilty.

Scenario 2:

- One of Alice, Bob, Carol, Dave, ... is the sender.
- Statistical analysis shows the following sender probabilities:
- Alice: Less than 1%
- Bob: Less than 1%
- Carol: Less than 40%
- Dave: Less than than 1%
- What will happen in practice?

Modes of Unlinkability

Classical Unlinkability: Entities exchange messages, we want unlinkability of any pair of

- **sender** of a message
- **reader** of a message
- **content** of a message

Distinguish from

- Who uses this service?
- Anonymous publishing only (writer-content unlinkability)
- Censorship free reading only (reader-content unlinkability)
- Content confidentiality (just encrypt)

Security Analysis

Needs for every solution:

1. Protection goals.
2. Attack model.
3. Attacker capabilities.

Typical attacks:

- Traffic analysis.
- Timing attacks.
- Side channel attacks.
- Active attacks.

Typical Solutions

High Latency Routing Obfuscation Solutions:

- Typical application: Email.
- **Disadvantages:** No interactivity due to high latency
- **Advantage:** Can be constructed very secure.

Low Latency Routing Obfuscation Solutions:

- Typical application: Web Services.
- **Advantage:** Convenient for real-time-near services.
- **Disadvantage:** Not very secure.

Other forms of approaches.

2. Superposed Sending

Charming protocol by David Chaum.

Anecdote of the dining cryptographers.

Cryptographical Anecdote

Anecdote of the Dining Cryptographers:

- Alice, Bob and Carol receive an invitation for dinner.
- The waiter informs them that the meal has been paid for.
- Alice, Bob and Carol want to find out if one of them or a third party has paid.
- Since the spender could be one of them, they want to keep his anonymity.

Centralized solution: A trusted entity.

- Assume the waiter is trusted.
- All privately tell the waiter.
- The waiter tells the result while keeping privacy guarantees.

Question: Is there a decentralized solution?

Decentralized Solution

Is an "anonymous broadcast communication" of one bit to all participants.

Also is a “secure multiparty computation” of a logical function of three inputs.

Preliminary Observation

- Every pair of nodes generates a 1-bit secret: s_{AB}, s_{CA}, s_{BC}.
- This secret is known to only these two nodes.
- Eg: A knows: s_{AB} and s_{CA}.
- Every node computes the xor of these two values she knows.
- Eg: A computes $s_{AB} \oplus s_{CA}$.
- Every node broadcasts the result to all other nodes.

Observation: In this case the number of 1 among the three broadcast bits is even.

- Equivalent: The xor of the three broadcast bits is 0.
- Equivalent: We have an invariant of 0 – independently from the specific situation.

Proof: $(s_{AB} \oplus s_{CA}) \oplus (s_{BC} \oplus s_{AB}) \oplus (s_{CA} \oplus s_{BC}) =$
$(s_{AB} \oplus s_{AB}) \oplus (s_{CA} \oplus s_{CA}) \oplus (s_{BC} \oplus s_{BC}) = 0 \oplus 0 \oplus 0 = 0$

Decentralized Protocol

Mechanism:

- Carry out the above protocol.
- If one of the three dinner guests paid,
  this person violates the described protocol by broadcasting the opposite result.

Interpretation:

- If the invariant still holds: NSA has paid.
- If the invariant is violated: One of them has paid.

Correctness of the result: Simple checking.

Analysis (1)

Let b_X be the bit broadcast by X: $b_A = s_{AB} \oplus s_{CA}$ $b_B = s_{AB} \oplus s_{BC}$ $b_C = s_{CA} \oplus s_{BC}$.

When nobody has paid
there are **even** 1s among the b.

Shared			Broadcast		
s_{AB}	s_{BC}	s_{CA}	b_A	b_B	b_C
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	0	0	0

When A has paid, it deviates
there are **odd** 1s among the b.

Shared			Broadcast		
s_{AB}	s_{BC}	s_{CA}	b_A	b_B	b_C
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	1	1	1
0	1	1	0	1	0
1	0	0	0	1	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	1	0	0

Analysis (2)

When nobody has paid
there are **even** 1s among the b.

Shared			Broadcast		
s_{AB}	s_{BC}	s_{CA}	b_A	b_B	b_C
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	1	1	1	0
1	0	0	1	1	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	0	0	0

When C has paid it deviates!
there are **odd** 1s among the b.

Shared			Broadcast		
s_{AB}	s_{BC}	s_{CA}	b_A	b_B	b_C
0	0	0	0	0	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	0	0	1

Analysis (3)

However B does not see s_{CA}. The two tables (a part from sorting of rows) look identical for B. B sees **that** one of A, C has paid but **not who**!

When A has paid as seen by B.

Shared			Broadcast		
s_{AB}	s_{BC}	s_{CA}	b_A	b_B	b_C
0	0		1	0	0
0	0		0	0	1
0	1		1	1	1
0	1		0	1	0
1	0		0	1	0
1	0		1	1	1
1	1		0	0	1
1	1		1	0	0

When C has paid as seen by B.

Shared			Broadcast		
s_{AB}	s_{BC}	s_{CA}	b_A	b_B	b_C
0	0		0	0	1
0	0		1	0	0
0	1		0	1	0
0	1		1	1	1
1	0		1	1	1
1	0		0	1	0
1	1		1	0	0
1	1		0	0	1

Analysis

Extension to **longer messages:**

- Extend protocol from 1 bit to n bits using rounds.
- In every round, one anonymous bit may be sent.
- Unconditionally secure protocol.
- Correct communication (provided in every round *at most one* party sends).
- Maintains privacy (unless all other participants collude).

Extension to **more participants:**

- Situation translates to n nodes with complete graph.
- Same result as with $n = 3$.
- Needs shared values on all $n \cdot (n-1)/2$ edges.

Using Sparse Graphs

Sparse Graph:

- Basically a similar situation.
- Topology dependent loss of some security properties.
- Linear scaling can be maintained at the price of security.

Example:

- Ring with secrets shared with left and right neighbor.
- If both neighbors conspire, privacy can be revoked.
- In complete graph all but one must conspire.

Collision Problem (1)

Problem:

- **Special** case: Only one or zero participants could adhere to the rule of "Behave differently if you have paid".
- **General** case: More than one party sends.
- Communication is disrupted by collisions.
- Similar to collisions in CSMA-type protocols.

Idea 1: Collision Prevention.

- Similar concept as with CSMA/CD.
- Detect collisions using checksums.
- In case of a collision, do an exponential backoff.
- May combine with protocol for reservations.
- Works only under the assumption of reasonable participants (honest but curious).
- Attacker can (anonymously) disrupt the network.

Collision Problem (2)

Idea 2: Trap Protocol: Catch the disrupter.

- Proposal for a (complex) protocol where an anonymous attacker can be caught.
- Was later broken: Can be used to break anonymity of honest participants.

Idea 3: Reservations Protocol.

- Provide a reservation protocol for participants.
- Participants must prove via zero knowledge protocol that they adhere to reservations.
- Quite complex, still unbroken.

3. Mix Networks

A **low latency solution**
for anonymous communication
with a touch of centralization.

Mix Network Scheme

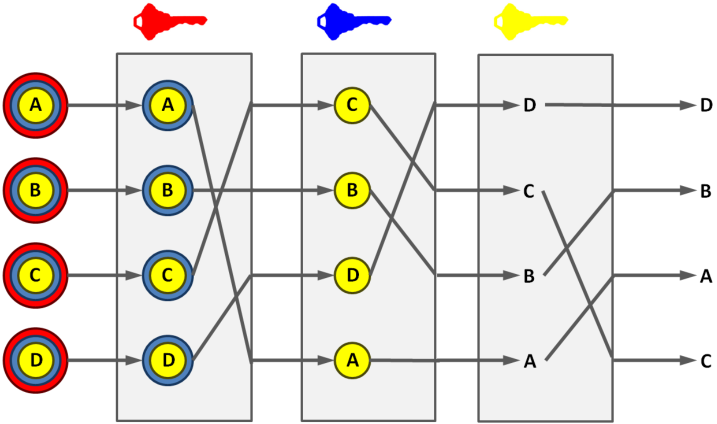

Fig. 1: A mix network © Rights see appendix.

Mix Network Operation

Mechanism:

- $n \geq 3$ nodes are operating in a linear cascade.
- Every node has a (public, private) key pair (e_i, d_i)
- Input into first node consists of an onion-like layer $e_1(e_2(e_3(m)))$.
- Every mix removes one layer of crypto and forwards to next node.

Attacks on Mixes (1)

Traffic analyst sees traffic between nodes and attempts to correlate traffic.

By sequence:

- First packet sent to first node corresponds to first packet received from last node.
- Prevent by reordering in the node.

By timing:

- Prevent by buffering messages for some time.
- Leads to (too) high latency.

Attacks on Mixes (2)

By content:

- Prevent by using (different) encryption from node to node.

By length:

- Prevent by sending only messages of one fixed length.

By number of messages:

- Prevent by sending decoy traffic.

Evaluation:

- Attacker cannot link sender and recipient.
- But: Attacker can identify participants in the system (from protocol handshake).
- But: Attacker can distinguish senders from recipients.

Plausible Deniability of Mix Use

Scenario 1: Use of tools for anonymous communication forbidden in some countries.

- Solution: Additional layers (tunnel, VPN or steganographic) hide handshake.

Scenario 2: Confirmation of suspicion

- Alice is suspect in a criminal case and her communication is intercepted.
- The day Alice learns that she is a suspect her use of mixing cascades goes up.
- This is no proof in court.
- This may trigger behavior of her observers.

General Recommendation

If you once in a while have to send something important with crypto grade security then always send with crypto grade security in order not to tip-off an attacker. Cryptographic and anonymous communication should be the default.

Problems with Mixes (1)

Problem: Collusion of Mixes

- A node should only know its own private key.
- How can this be guaranteed
  when an entire cascade is operated by a single privacy-service?
- Idea: The individual nodes should be organizationally independent.

Problem: Authenticity of Mixes

- **Attack:** Set up an anonymizer only to catch interesting information
- **Question:** How to distinguish true from fake anonymization service?
- **Question:** Why should I trust a security service more than a possible attacker?
  Just because they call themselves security service?
  Or rather because I have means to verify trust aspects!

Problems with Mixes (2)

Problem: Scaling

- Security gets better when more and independent nodes use the system
- Thought experiment 1: Only 1 node uses the system.
- Thought experiment 2: Only 2 nodes use the system.
- Thought experiment 3: 1 node plus 500 nodes of the NSA use the system.
- Thought experiment 4: 100 different nodes plus 500 nodes of the NSA use the system.

Problem: Collusion of Other Users

- If **all** the other users conspire against me, anonymity can be broken easily.

JAP and AN.ON

JAP and AN.ON

- Initiated by TU Dresden and
  Unabhängiges Landeszentrum für den Datenschutz Schleswig-Holstein.
- Fixed cascade of three nodes.
- All nodes operated by well-known entities.
- User can chose from several cascades.
- More Information

4. Remailers

High latency solutions
for anonymous communication.

Overview

Overview:

- First attempt to develop working anonymous communication.
- Several conceptually interesting development steps.
- Today mostly defunct and superseded by other, by low latency tech (TOR, I2P).
- Sad: High latency remailers would offer much better anonymity than low latency tech.

Timing / Flow attack:

- Attacker watches packets flow between nodes.
- Attacker produces correlations between traffic.
- With low latency (3s end-to-end) this is rather easy.
- High latency does a store-reschuffle sequence-forward approach for several days.
- Problem: If only 2, 3 people use it – the anonymity set is too small.
  The convenience of the many (using low latency tech) produces the risk for all.

4 Types of Classical Remailers

4 Types of classical remailers

- Type 0: Pseudonymous Remailers
- Type 1: Cypherpunk Remailers
- Type 2: Mixmaster Remailers
- Type 3: Mixminion Remailers

Type 0: Pseudonymous Remailers (1)

Idea: First attempt at remailers: anon.penet.fi by Johan Helsingius.

Mechanism:

- Sender provides email address and registers a pseudonym.
- Sender sends mail to remailer.
- Remailer removes identifying headers.
- Remailer fills in pseudonymous address.
- Remailer forwards to final recipient.
- Receiver replies to pseudonymous address.
- Remailer forwards in similar fashion.

Type 0: Pseudonymous Remailers (2)

Analysis: Many problems.

- Remailer knows original addresses and address mappings.
- No security against attacks from remailer itself.
- Remailer can be compromised or subpoenaed.
- Susceptible to eavesdropping attacks since messages are sent as plain text.
  But: User can use payload encryption.
- Susceptible to traffic analysis attacks.
- Susceptible to replay attacks.

Type 1: Cypherpunk Remailers

Idea: Partially solve problem of plain text transport by encryption.

Mechanism:

- User retrieves public key of remailer.
- User sends encrypted message to remailer with an additional Anon-To header indicating true recipient
- Remailer decrypts
- Remailer removes identifying information
- Remailer forwards to true recipient in Anon-To header.

Analysis:

- Secure against eavesdropping by third parties.
- Susceptible against eavesdropping by remailer; user can employ separate encryption.
- No reply possible.
- Remailer knows sender – but can use chains of remailers.
- Susceptible to traffic analysis and replay attacks.

Type 2: Mixmaster Remailers

Idea: Solve problem of traffic analysis by mixing.

Mechanism: First application of mix concept.

Analysis:

- No reply possible
- High latency allows excellent security.
- Body may describe a reverse path, but no automatic protocol provided mechanism
- Replay attacks possible

Type 3: Mixminion Remailers (1)

Idea: Solves most remaining problems of remailers.

Design document by the inventors of the concept nicely illustrates the many important aspects of anonymous communication.

Concept: Single Use Reply Block (SURB)

- Along the path of mail delivery, encode and encrypt a layered return path.
- Receiver of the message may reply but does not learn identity of partner.

Concept: Preventing replay attacks by key rotation

- Problem: Do not want to have time stamps (could allow attacks).
- Problem: Do not want to have serial numbers (need to keep status, which is operational burden and could allow attacks).
- Solution: Use changing encryption keys.

Type 3: Mixminion Remailers (2)

Concept: Dummy traffic.

- When volume of traffic is too low, traffic analysis may succeed.
- Remailers generate dummy traffic to prevent traffic analysis.

Concept: Spam prevention via exit policies

- Every anonymously delivered mail comes with instructions how recipient can confidentially request not to get more anonymous mail from a remailer.

Analysis:

- Great concept, currently mostly defunct.
- More information available: Active (?) github Original github

Other Mail Services

Anonymous mailing services on top of other (mostly low latency) technologies:

- I2PBote
- BitMessage
- TorMail (now defunct)

5. Onion Routing

A **low latency** solution
for anonymous communication
with strong distribution.

Tor Basics

Idea: A kind of distributed, decentralized mix cascade.

Three types of nodes

1. **Guard** node: Knows identity of the Tor network user.
2. **Relay** node: Knows only guard and exit node.
3. **Exit** node: Knows the relay node and the resource which is accessed.

TOR Circuit:

- Anonymous replacement for TCP protocol.
- First set up Tor circuit.
- Then use circuit for the remainder of the session.
- Normal Tor circuit uses 3 nodes.

How Tor Works (1)

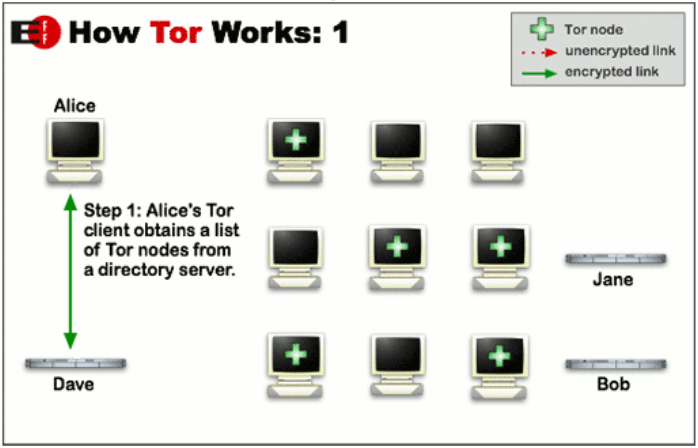

Fig. 2: Alice contacts the directory server to obtain a list of Tor nodes. © Rights see appendix.

How Tor Works (2)

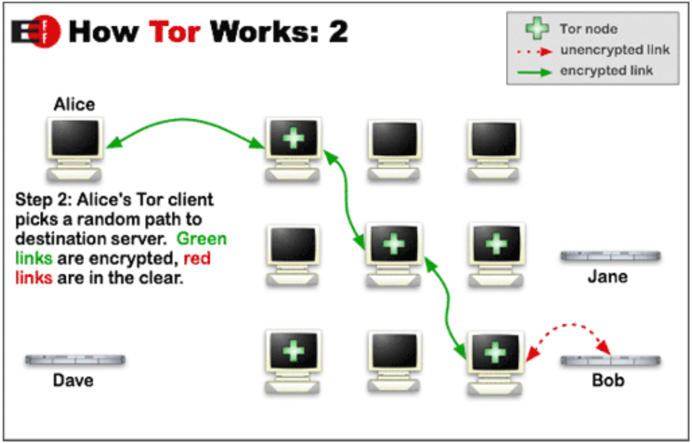

Fig. 3: Alice builds up a Tor circuit to the node she uses as exit node. © Rights see appendix.

How Tor Works?

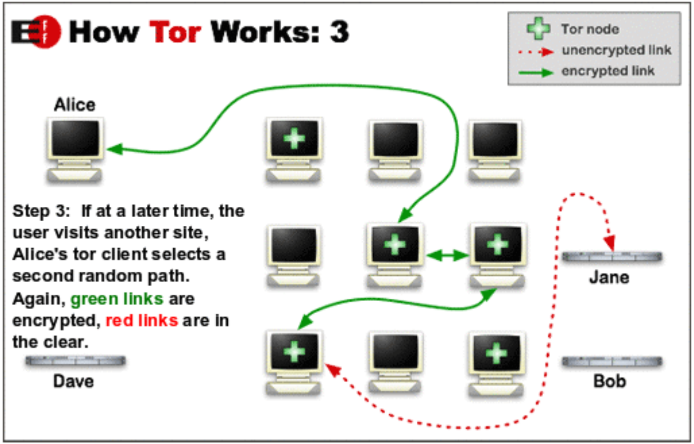

Fig. 4: Alice uses Tor at another occasion. © Rights see appendix.

Attacks Against Tor

Attack Scenarios:

- Attacker controls all three nodes: Can link surfer to website.
- Attacker controls guard & exit: Timing and packet number attack on guard & exit.

Important:

- Chose the right guard, since the guard knows who you are.
- **Variant 1:** Chose a trusted guard.
- **Variant 2:** Next best option: Chose a random guard once in a while.

Practical Use of Tor

Compromises:

- Tor is an operative system which requires compromises of performance and anonymity.
- Tor does not use padding; some mild padding was introduced recently.
- Tor does not use decoy traffic.
- Tor only transports TCP.
  Negative: For example, VoIP or DNS over Tor does not work.
  Positive: Other protocols could leak identity information.

Riscs in operating an exit node:

- Forwarding requests to dubious sites.
- Seizing of equipment and legal trouble.
- Attention of three-letter-agencies.

Map of Tor Relais

Fig. 5: This map of Tor relais nodes shows that operating a normal relais node is quite popular. © Rights see appendix.

Map of Tor Exit Nodes

Fig. 6: Map of Tor exit nodes shows that operating exit nodes is less common in countries known for more restrictive legal systems. © Rights see appendix.

Can We Trust Tor?

Basic evaluation:

- Open source project.
- Active research on Tor security.
- Some centralized components: Directory server.
- Many decentralized components: Nodes.

Yes, provided:

- We know a lot about Tor.
- We follow the pertinent research.
- We adhere to the (many) security rules.
- We do not operate services drawing in focused attacks.

No, provided:

- We assume the existence of a global traffic analyst.
- We need interactive, responsive Web 2.0 convenience.
- We operate out of Tor-banning countries.

Nym Situation in TOR

Remailer Anonymity:

- Attacker knows the email addresses of all receivers.
- Attacker knows the email addresses of all sender.
- Attacker cannot link a specific sender to a specific receiver.

TOR Anonymity:

- Attacker knows the IP address of surfers.
- Attacker knows the IP address of servers.
- Attacker cannot link a specific surfer to a specific server.

TOR Hidden Service Anonymity:

- Attacker knows the IP address of surfers.
- Attacker does not know the IP address of a hidden service.
- Attacker cannot link a specific surfer to a specific server.
- Attacker cannot link a hidden service to a person.

What are Hidden Services?

Paradoxical Situation:

- **Naming:** Surfer uses (names, references) a service without knowing its IP address.
- **Routing:** Surfer routes to a service. without having or compromising its IP address.

Answers:

- Use `.onion` addresses for naming.
- Use an untraceable routing mechanism
- Note: Tor exit nodes are known to attackers and cannot serve as service providers.

Hidden Services (1)

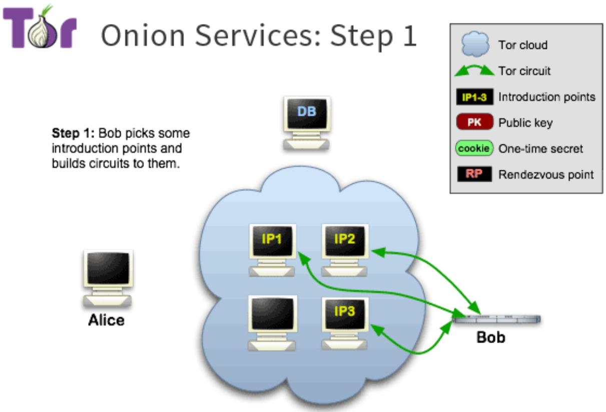

Fig. 7: Hidden Services (1) © Rights see appendix.

Hidden Services (2)

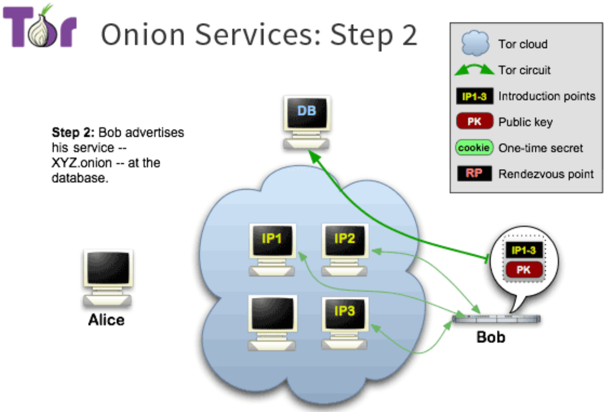

Fig. 8: Hidden Services (2) © Rights see appendix.

Hidden Services (3)

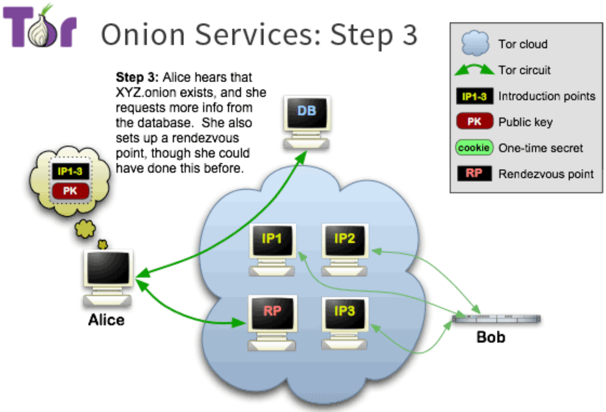

Fig. 9: Hidden Services (3) © Rights see appendix.

Hidden Services (4)

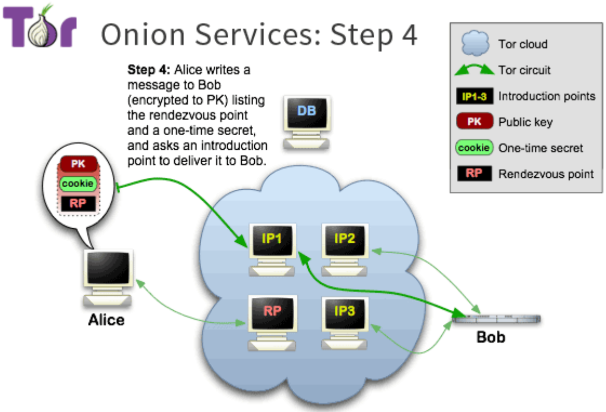

Fig. 10: Hidden Services (4) © Rights see appendix.

Hidden Services (5)

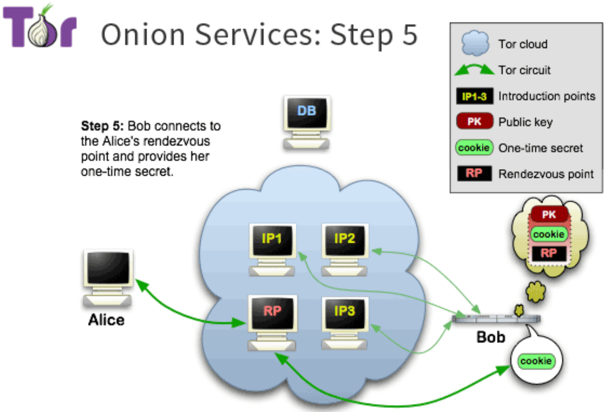

Fig. 11: Hidden Services (5) © Rights see appendix.

Hidden Services (6)

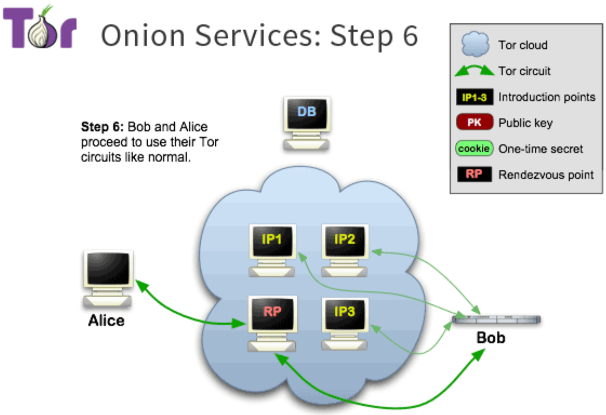

Fig. 12: Hidden Services (6) © Rights see appendix.

Analysis of Hidden Services

Purposes are often illegal

- Botnet command and control servers
- Drug, weapon, illegal goods sale
- Ongoing debate how to ban illegality without compromising anonymity.

Problem 1: Attacks.

- Traffic correlation & side channel attacks can deanonymize hidden services.

Problem 2: Trust

- There is no trust / reputation source, so you can end up at fake sites.

I2P

Comparison:

- Many conceptual similarities with Tor.
- More advanced and flexible than Tor.
- Smaller community with less funding, less activity, smaller anonymity set.

Two Essential Differences:

- **Garlic routing** encrypts several payload messages into message.
  Tracking is more difficult than with onion routing.
- **Unidirectional tunnels** instead of bidirectional tunnels as with Tor.

6. Further Remarks

Another solution
and some further problems.

Dolev Bus

Description in a paper of Beimel and Dolev.

Mechanism:

- Every user is a bus station.
- All bus stations from a ring.
- There is a bus going around the ring.
- At every bus stations messages may “hop on” or “get off” the bus.
- Encryption from station to station for every passenger seat prevents tracking.
- Constant size of the bus prevents length correlation.

Variants:

- Use a second bus going in the opposite direction.
- Use different topologies and bus schedules.

Problems

Wide range of practical problems must be solved:

- Identity leaks via browser fingerprinting, cookies, DNS traffic, Javascript snippets, ...
  Tor developers recommend use of special Tor browser bundle.
- Stupid user leaks identity via content ("Yours sincelery, Tom Sawyer").
- User uses unencrypted services and exit node can intercept.
- Javascript picks up usage characteristics (keyboard typing is a biometric signal!).
  Tor browser should have Javascript turned off.
- User leaks identity via writing style: Paper
- High security requirements may damage web surfing quality.

The practice of really secure anonymous communication is difficult.

Broken Services

Die folgende Tabelle zeigt eine Liste bekannter Webproxys, die den Anonymitätstest der JonDos GmbH nicht bestehen:

Betreiber	HTML/CSS/FTP	JavaScript	Java
Anonymouse	Gebrochen	Gebrochen	Gebrochen
Cyberghost Web	-	Gebrochen	Gebrochen
Hide My Ass!	-	Gebrochen	Gebrochen
WebProxy.ca	-	Gebrochen	Gebrochen
KProxy	Gebrochen	Gebrochen	Gebrochen
Guardster	-	Gebrochen	Gebrochen
Megaproxy	Gebrochen	(kostenfrei nicht verfügbar)	(kostenfrei nicht verfügbar)
Proxify	-	Gebrochen	Gebrochen
Ebumna	Gebrochen	Gebrochen	Gebrochen

Fig. 13: A very large number of self-proclaimed anonymization services are broken.

Appendix

List of Figures

List of Rights (1/2)

Fig. 1 Source: https://commons.wikimedia.org/wiki/File:Red_de_mezcla.png Primepq by CC BY-SA 3.0 https://creativecommons.org/licenses/by-sa/3.0

Fig. 2 Electronic Frontier Foundation, CC BY 3.0 https://creativecommons.org/licenses/by/3.0

Fig. 3 Electronic Frontier Foundation, CC BY 3.0 https://creativecommons.org/licenses/by/3.0

Fig. 4 Electronic Frontier Foundation, CC BY 3.0 https://creativecommons.org/licenses/by/3.0

Fig. 5 Source: https://tormap.void.gr/, Screenshot 2018.

Fig. 6 Source: https://tormap.void.gr/, Screenshot 2018.

Fig. 7 Source: https://2019.www.torproject.org/docs/onion-services

Fig. 8 Source: https://2019.www.torproject.org/docs/onion-services

Fig. 9 Source: https://2019.www.torproject.org/docs/onion-services

Fig. 10 Source: https://2019.www.torproject.org/docs/onion-services

Fig. 11 Source: https://2019.www.torproject.org/docs/onion-services

Fig. 12 Source: https://2019.www.torproject.org/docs/onion-services

Fig. 13 Source: https://www.privacy-handbuch.de/handbuch_22b2.htm

Terms of Use (1)

Terms of Use (2)

Disclaimer: Die sich immer wieder ändernde Rechtslage für digitale Urheberrechte erzeugt für mich ein nicht unerhebliches Risiko bei der Einbindung von Materialien, deren Status ich nicht oder nur mit unverhältnismäßig hohem Aufwand abklären kann. Ebenso kann ich den Rechteinhabern nicht auf sinnvolle oder einfache Weise ein Honorar zukommen lassen, obwohl ich – und in letzter Konsequenz Sie als Leser – ihre Leistungen nutzen.

Daher binde ich gelegentlich Inhalte nur als Link und nicht durch Framing ein. Lt EuGH Urteil 13.02.2014, C-466/12 ist das unbedenklich, da die benutzten Links ohne Umgehung technischer Sperren auf im Internet frei verfügbare Inhalte verweisen.

Wenn Sie diese Rechtslage stört, dann setzen Sie sich für eine Modernisierung des völlig veralteten Vergütungssystems für urheberrechtliche Leistungen ein. Bis dahin klicken Sie bitte auf die angegebenen Links und denken Sie darüber nach, warum wir keine für das digitale Zeitalter sinnvoll angepaßte Vergütungssysteme digital erbrachter Leistungen haben.

Zu Risiken und Nebenwirkungen fragen Sie Ihren Rechtsanwalt oder Gesetzgeber.

Weitere Hinweise finden Sie im Netz hier und hier oder hier.

Citing This Document

If you use contents from this document or want to cite it,
please do so in the following manner:

Clemens H. Cap: Anonymous Communication. Electronic document. https://iuk.one/1033-1013 3. 7. 2021.

Bibtex Information: https://iuk.one/1033-1013.bib

```
@misc{doc:1033-1013,
    author      = {Clemens H. Cap},
    title       = {Anonymous Communication},
    year        = {2021},
    month       = {7},
    howpublished = {Electronic document},
    url         = {https://iuk.one/1033-1013}
}
```

Typographic Information:
Typeset on July 3, 2021
This is pdfTeX, Version 3.14159265-2.6-1.40.21 (TeX Live 2020) kpathsea version 6.3.2
This is pgf in version 3.1.5b
This is preamble-slides.tex myFormat©C.H.Cap

List of Slides

Legend:
- continuation slide
- slide without title header
- image slide