Anonymity On The Web

Francesco Locatello
Michael König ETH Zürich

April 29, 2015
Fact:

We are being watched.
Who needs anonymity?

Normal people:

- Identity thieves
- Irresponsible corporations
- Sensitive topics
- Circumvent censorship
Who needs anonymity?

- Identity thieves
- Irresponsible corporations
- Sensitive topics
- Circumvent censorship
Who needs anonymity?

Normal people:
- Identity thieves
Who needs anonymity?

Normal people:
- Identity thieves
- Irresponsible corporations
Who needs anonymity?

Normal people:
- Identity thieves
- Irresponsible corporations
- Sensitive topics
Who needs anonymity?

Normal people:
- Identity thieves
- Irresponsible corporations
- Sensitive topics
- Circumvent censorship
How to retain it?

Tor mission:
“Tor aims to provide protection for ordinary people who want to follow the law.”
How to retain it?

Tor mission: "Tor aims to provide protection for ordinary people who want to follow the law."
What to do with Tor:

- Access web sites anonymously
- Host web servers with anonymous location
What to do with Tor:

- Access web sites anonymously
What to do with Tor:

- Access web sites anonymously
- Host web servers with anonymous location
Tor in real life
Tor in real life

Torbook helps you connect and share with the people in your life.

Sign Up
It's free and always will be.

* Full Name:

* Email:

* Password:

Captcha Challenge
r e e j 9

Type in the verification code above

Sign Up
Tor in real life

Silk Road
anonymous market

Shop by Category

Drugs 8,670
Cannabis 2,066
Dissociatives 165
Ecstasy 660
Opioids 591
Other 455
Precursors 50
Prescription 2,146
Psychedelics 981
 Stimulants 1,102

Apparel 264
Art 127
Biotic materials 1
Books 881
Collectibles 5
Computer equipment 32
Custom Orders 66
Digital goods 509
Drug paraphernalia 305
Electronics 77

messages 0 | orders 0 | account $0.00

Search

1g MDMA 82%+ High Quality - Made in Germany - $1.30
50 gr. Crystal MDMA Rocks - $23.33
Valium 10mg/ Diazepam (100 Pills) - $2.32
3g XxX AAA QUALITY WEED, AMAZING - $0.98

Kamagra jelly (India), 1 week pack - $0.98
Honeycomb Wax (85% THC) Fully Purged - $1.45
1 gram - Moroccan Hash - DUTCH QUALITY - $0.27
Citalopram 10x 20mg table - $0.10
Anonymity On The Web

Definition:
Allow users to communicate privately by hiding their identities from the recipient or third parties on the internet.
Anonymity On The Web

Definition:
Allow users to communicate privately by hiding their identities from the recipient or third parties on the internet.
A web prospective
A web prospective

The web cloud
A web prospective

Direct connection

192.0.2.0
A web prospective

Tor breaks this link
A web prospective

Host website anonymously: no registered domain name, no hosting account
Outline

1 Tor
   - Structure
   - Strengths
   - Weaknesses

2 Dissent
   - Foundations of Anonymity

3 Conclusion
How to use Tor:
Download the Tor client also called Onion proxy
How to use Tor:
Download the Tor client also called Onion proxy

What does Tor do for you:
Tor protects the transport of data, it doesn’t hide user informations (Tor browser).
Getting started with anonymity
Getting started with anonymity

Hey, why don’t we watch a movie after your presentation?
Getting started with anonymity
Getting started with anonymity

You always put yourself before our love, that's too much! I don't wanna see you ever again.
Getting started with anonymity
Proxy

Do you trust the proxy?

You
IP: 1.1.1.1

Proxy server
This can cache content, do webfiltering, or just make you anonymous
IP 2.2.2.2

Destination
This website will think you are IP: 2.2.2.2 making you anonymous
Do you trust the proxy?
The topology of the Tor Network

- Tor client
- Entry guard
- Tor network
- Middle relay
- Exit relay
- Destination

- Encrypted by Tor
- Not encrypted by Tor
The topology of the Tor Network

- Ran by volunteers all over the world
The topology of the Tor Network

- Ran by volunteers all over the world
- Learning what sites you visit
The topology of the Tor Network

- Ran by volunteers all over the world
- Learning what sites you visit
- Learning your location
The Onion Routing
Connection Scheme

How Tor Works

Alice’s Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.

Tor node

unencrypted link
encrypted link

Alice

Jane

Bob
Performance: Latency and Bandwidth
Performance: Latency and Bandwidth

How Tor Works

Alice's Tor client picks a random path to destination server. **Green links are encrypted, red links are in the clear.**
Outline

Possible Attacks:
- Side channel analysis introduction
  - Global traffic analysis (1)
  - Active attack: congestion (2)
- Intersection attack (3)
- Software exploitation and self identification (4)
See both sides of a communication channel
See both sides of a communication channel

\[ c = \# \text{ of controlled relays} \]
\[ n = \# \text{ of relays} \]
See both sides of a communication channel

$c = \# \text{ of controlled relays}$

$n = \# \text{ of relays}$

$\downarrow$

$\text{correlation of traffic with } p = \text{???}$
See both sides of a communication channel

\[ c = \# \text{ of controlled relays} \]
\[ n = \# \text{ of relays} \]

\[ p = \frac{c}{n} \text{ correlation of traffic with} \]
Side Traffic Attack

Execution Analysis

- Break cryptography
Side Traffic Attack

Execution Analysis
- Break cryptography

Traffic Analysis
- Correlate time and size of packets
## Side Traffic Attack

### Execution Analysis
- Break cryptography

### Traffic Analysis
- Correlate time and size of packets
- Deduce the path through the network
Global Traffic Analysis on Tor (1)
Global Traffic Analysis on Tor (1)
A Simple Example
A Simple Example
A Simple Example
How Tor handles it:
How Tor handles it:
Why entry guards:

Those relays are not controlled or observed.

Those relays are observed or controlled.
Why entry guards:

Those relays are not controlled or observed
Why entry guards:

- Those relays are not controlled or observed
- Those relays are observed or controlled
Explanation: analysis over a month

Probability being safe with entry guards:

$$p = (1 - c)^n$$

Probability being safe without entry guards:

$$p_{\text{all safe}} = p_{\text{number of connections safe}} = 0$$ for number of connections sufficiently big.
Explanation: analysis over a month

Probability being safe with entry guards: \( p = (1 - \frac{c}{n})^3 \)
Probability being safe with entry guards: \( p = (1 - \frac{c}{n})^3 \)

Probability being safe without entry guards:

\[
p_{\text{all safe}} = p_{\text{safe}}^\text{number of connections} = 0
\]

for number of connection sufficiently big.
Active Attack: Congestion (2)

Assumptions:
The attacker can either be "in the network" or own or have compromised a web server
The attacker wishes to determine the set of relays through which a long lived circuit owned by a particular user passes (SSH).
Active Attack: Congestion (2)

Assumptions:
Active Attack: Congestion (2)

Assumptions:

The attacker can either be "in the network" or own or have compromised a web server
Assumptions:

The attacker can either be "in the network" or own or have compromised a web server

The attacker wishes to determine the set of relays through which a long lived circuit owned by a particular user passes (SSH).
Strategy

*Induce heavy load to cause congestion and forwarding delays*

Attack Client

Victim Client

See which emerging flows are affected

Public Server
Intersection Attack: framework (3)
Intersection Attack: framework (3)

One time interaction are rare
Intersection Attack: framework (3)
Intersection Attack: framework (3)
Effectiveness
Harvard Student Receives F For Tor Failure While Sending 'Anonymous' Bomb Threat

On Tuesday, the FBI filed a criminal complaint against a Harvard University sophomore student for making bomb threats that led school officials to delay some final exams, including his, that had been scheduled for Monday. According to the five-page complaint, the student “took steps to disguise his identity” by using Tor, a software which allows users to browse the web anonymously, and Guerrilla Mail, a service which allows users to create free, temporary email addresses.
Documents obtained by The Washington Post indicate that the National Security Agency is collecting billions of records a day to track the location of mobile phone users around the world. This bulk collection, performed under the NSA’s international surveillance authority, taps into the telephony links of major telecommunications providers including some here in the United States.

The NSA collects this location and travel habit data to do “target development” — to find unknown associates of targets it already knows about.

To accomplish this, the NSA compiles information on a vast database of devices and their locations. Most of those collected, by definition, are suspected of no wrongdoing. Officials say they do not purposely collect U.S. phone locations in bulk, but a large number are swept up “incidentally.”

Using these vast location databases, the NSA applies sophisticated analytics techniques to identify what it calls co-travelers — unknown associates who might be traveling with, or meeting up with a known target.

HERE IS HOW IT WORKS
Software Exploits and Self Identification (4)
Software Exploits and Self Identification (4)
Wrap up

<table>
<thead>
<tr>
<th>Attack</th>
<th>Tor</th>
<th>Dissent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Traffic analysis (1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Congestion attack (2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intersection attack (3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software exploits (4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Wrap up

<table>
<thead>
<tr>
<th>Attack</th>
<th>Tor</th>
<th>Dissent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Traffic analysis (1)</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Congestion attack (2)</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Intersection attack (3)</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Software exploits (4)</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Dissent: Introduction
Alternative foundation for anonymity:
Alternative foundation for anonymity:

- Verifiable shuffles
Alternative foundation for anonymity:
- Verifiable shuffles
- Dining cryptographers
Dissent: Introduction

Alternative foundation for anonymity:
- Verifiable shuffles
- Dining cryptographers

Framework:
Dissent: Introduction

Alternative foundation for anonymity:
- Verifiable shuffles
- Dining cryptographers

Framework:
- A group of users wants to share secrets between themselves
Verifiable Shuffles: Mixing Server
Verifiable Shuffles: Mixing Server

Sender 1

\[ m_1 \]

... \[ \cdots \] ...

\[ m_\pi(1) \]

Mixer \[ \pi \]

\[ m_\pi(n) \]

Sender n

\[ m_n \]

mix-servers
System Overview

\[ m_1 \quad \ldots \quad m_n \]

\[ E(m_1) \quad \ldots \quad E(m_n) \]

\[ \text{Mix-net} \]

\[ E'(m_{\pi(1)}) \quad \ldots \quad E'(m_{\pi(n)}) \]

\[ \text{Decryption} \]

\[ m_{\pi(1)} \quad \ldots \quad m_{\pi(n)} \]

senders

mix-servers
Mixing Network

Mix-server 1 $\pi_1$

$E(m_1) \xrightarrow{\pi_1} E'(m_{\pi_1(1)}) \xrightarrow{\pi_1} \cdots \xrightarrow{\pi_1} E''''(m_{\pi_1(n)})$

Mix-server N $\pi_N$

$E(m_1) \xrightarrow{\pi_N} E'(m_{\pi_N(1)}) \xrightarrow{\pi_N} \cdots \xrightarrow{\pi_N} E''''(m_{\pi_N(n)})$

$\pi = \pi_N \circ \cdots \circ \pi_1$
Mixing Network

Synchronous round: concentric layers of public key encryption

\[ \pi = \pi_N \circ \ldots \circ \pi_1 \]
Mixing Network

Synchronous round: concentric layers of public key encryption

Each shuffler: unwraps, permutes and forwards

\[ \pi = \pi_N \circ \ldots \circ \pi_1 \]
Mixing Network

- Synchronous round: concentric layers of public key encryption
- Each shuffler: unwraps, permutes and forwards
- The final shuffler: broadcasts

\[ \pi = \pi_N \circ \ldots \circ \pi_1 \]
Considerations

Provable anonymity

Worst possible traffic at each shuffler

Practical only when high latencies are tolerable
Considerations

- Provable anonymity
Considerations

- Provable anonymity
- *Worst possible traffic* at each shuffler
Considerations

- Provable anonymity
- **Worst possible traffic** at each shuffler
- Practical only when high latencies are tolerable
Dining cryptographers

The only well studied foundation for anonymity not based on sequential relaying is Dining Cryptographers or DC-nets.
Dining cryptographers
Considerations

A slow or offline member requires restart from scratch

Any malicious member can jam with random bits
Considerations

A slow or offline member requires restart from scratch

Any malicious member can jam with random bits
Tradeoff

Weak anonymity among many nodes via onion routing
Tradeoff

Weak anonymity among many nodes via onion routing

Strong anonymity among few nodes with DC-nets
Extension

- Client/server architecture
Client/server architecture

Clients trust only that at least one server in the set is honest, but need not know or choose which server to trust
Dissent Protocol Outline Setup
Clients A, B, C each submit pseudonym signing key, which the servers shuffle.

Shuffle output order determines clients' transmission order in DC-net exchanges.
Round Structure
Round Structure

Client A: $S_{A,X} \oplus S_{A,Y}$

Client B: $S_{B,X} \oplus S_{B,Y}$

Client C: Offline

Client D: $S_{D,X} \oplus S_{D,Y}$

Server X: $S_{A,X} \oplus S_{B,X} \oplus S_{C,X} \oplus S_{D,X}$

Server Y: $S_{A,Y} \oplus S_{B,Y} \oplus S_{C,Y} \oplus S_{D,Y}$

Plaintext Output: msg 1, msg 3

Exchange $i$:
- Slot 2 (closed)
- Slot 4 (closed)
Scalability

Client: shares secrets with only $M < N$ servers

Client: compute $M$ pseudo-random bits per clear text bit

Server: compute $N$ pseudo-random bits per clear text bit

Parallelizable computation

Network churns tolerance
Client: shares secrets with only $M \ll N$ servers
Client: shares secrets with only $M << N$ servers
Client: compute $M$ pseudo-random bits per clear text bit
Scalability

- Client: shares secrets with only $M << N$ servers
- Client: compute $M$ pseudo-random bits per clear text bit
- Server: compute $N$ pseudo-random bits per clear text bit
Scalability

- Client: shares secrets with only \( M \ll N \) servers
- Client: compute \( M \) pseudo-random bits per clear text bit
- Server: compute \( N \) pseudo-random bits per clear text bit
- Parallelizable computation
Scalability

- Client: shares secrets with only $M \ll N$ servers
- Client: compute $M$ pseudo-random bits per clear text bit
- Server: compute $N$ pseudo-random bits per clear text bit
- Parallelizable computation
- Network churns tolerance
Handling attacks

(a) Onion routing is vulnerable to passive and active fingerprinting attacks.

(b) Cascade mixes or verifiable shuffles collectively “scrub” traffic patterns.
## Attacks Comparison

<table>
<thead>
<tr>
<th>Attack</th>
<th>Tor</th>
<th>Dissent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Traffic analysis (1)</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Congestion attack (2)</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Intersection attack (3)</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Software exploits (4)</td>
<td></td>
<td>✗</td>
</tr>
</tbody>
</table>
## Attacks Comparison

<table>
<thead>
<tr>
<th>Attack</th>
<th>Tor</th>
<th>Dissent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Traffic analysis (1)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Congestion attack (2)</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Intersection attack (3)</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Software exploits (4)</td>
<td>✗</td>
<td></td>
</tr>
</tbody>
</table>
## Attacks Comparison

<table>
<thead>
<tr>
<th>Attack</th>
<th>Tor</th>
<th>Dissent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Traffic analysis (1)</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Congestion attack (2)</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Intersection attack (3)</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Software exploits (4)</td>
<td>✗</td>
<td></td>
</tr>
</tbody>
</table>
## Attacks Comparison

<table>
<thead>
<tr>
<th>Attack</th>
<th>Tor</th>
<th>Dissent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Traffic analysis (1)</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Congestion attack (2)</td>
<td>✗</td>
<td>✔️</td>
</tr>
<tr>
<td>Intersection attack (3)</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Software exploits (4)</td>
<td>✗</td>
<td></td>
</tr>
</tbody>
</table>
# Attacks Comparison

<table>
<thead>
<tr>
<th>Attack</th>
<th>Tor</th>
<th>Dissent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Traffic analysis (1)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Congestion attack (2)</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Intersection attack (3)</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Software exploits (4)</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Limitations

- Scalability still limited
- Intersection attacks
- Handling server failure
Latency Considerations

![Graph showing time (seconds) to download page vs. size (bytes) of all index page content (HTML page, images, JS, CSS) with different latency characteristics: No Anonymity, Tor, Dissent, Dissent + Tor.]
Wrap up

- Latency security tradeoff for the transport of the data
Wrap up

- Latency security tradeoff for the transport of the data
  - Low latency: Tor
    - Weak anonymity guarantees
Wrap up

- Latency security tradeoff for the transport of the data
  - Low latency: Tor
    - Weak anonymity guarantees
  - Strong anonymity: Dissent
    - High latency
Attacks against anonymity can be done at multiple levels
Conclusion

- Attacks against anonymity can be done at multiple levels
- There are no out of the box solutions, but....
Conclusion

- Attacks against anonymity can be done at multiple levels
- There are no out of the box solutions, but....
- There exist a set of tools that can help to provide the required level of anonymity (Tor, Tor Browser, VM, Dissent).
Questions?