Privacy at the communication layer

Seeing Through Network-Protocol Obfuscation
Wang, Dyer, Akella, Ristenpart, and Shrimpton 2015

CS-721

Carmela Troncoso
http://carmelatroncoso.com/
Beyond anonymity: Censorship prevention

Adversary's goal: prevent communication between two parties

2-step process:

Finding the flow: fingerprinting

Prevent communication: direct censor
FINDING THE FLOW: FINGERPRINTING

DESTINATION:
IP addresses, hosts, ports,… Tor (or other anon comm)

CONTENT:
protocol-strings, keywords, domains, http hosts,… Encryption

FLOW PROPERTIES:
length, inter-arrival times, bursts, Obfuscation, mimic

PROTOCOL SEMANTICS:
protocol behavior (mostly active attacks)

FINDING THE FLOW: FINGERPRINTING

Destination:
IP addresses, hosts, ports,...  Tor (or other anon comm)

Content:
protocol-strings, keywords, domains, http hosts,...  Encryption

Flow properties:
length, inter-arrival times, bursts, Obfuscation, mimic

Protocol semantics:
protocol behavior (mostly active attacks)

Sheharbano Khattak, Tariq Elahi, Laurent Simon, Colleen M. Swanson, Steven J. Murdoch, and Ian Goldberg.
**ScrambleSuit**

**Pseudo-random payload:** ScrambleSuit computationally indistinguishable from randomness. I.e., no DPI fingerprints.

![Figure 1: ScrambleSuit’s protocol stack.](image-url)
**ScrambleSuit**

**Pseudo-random payload:** ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

**Polymorphic:** changes shape to hinder classification.

![Figure 1: ScrambleSuit's protocol stack.](image)
**ScrambleSuit**

**Pseudo-random payload:** ScrambleSuit computationally indistinguishable from randomness, i.e., no DPI fingerprints.

**Polymorphic:** changes shape to hinder classification.

**Usable:** integrated in Tor & moderate overhead.

---

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
ScrambleSuit

**Pseudo-random payload:** ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

**Polymorphic:** changes shape to hinder classification.

**Usable:** integrated in Tor & moderate overhead.

**Defense against active probing:**

Figure 1: ScrambleSuit’s protocol stack.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
ScrambleSuit

**Pseudo-random payload:** ScrambleSuit computationally indistinguishable from randomness, i.e., no DPI fingerprints.

**Polymorphic:** changes shape to hinder classification.

**Usable:** integrated in Tor & moderate overhead.

**Defense against active probing:**

Ticket system?

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
**ScrambleSuit**

**Pseudo-random payload**: ScrambleSuit computationally indistinguishable from randomness. i.e., no DPI fingerprints.

**Polymorphic**: changes shape to hinder classification.

**Usable**: integrated in Tor & moderate overhead.

**Defense against active probing**: use of a secret which is shared between client and server and exchanged out-of-band.

---

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
ScrambleSuit: defending against active probing

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
ScrambleSuit

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
ScrambleSuit: Shaping

Shaping approach:

**PROTOCOL POLYMORPHISM:** one protocol shape for every server

**PACKET LENGTHS and INTER-ARRIVAL TIMES**

**ON BOOTSTRAPPING:**

generates a 256-bit seed to obtain two discrete probability distributions
seed transmitted to clients so that it is two-way
It is difficult to evaluate the effectiveness of our obfuscation techniques since ScrambleSuit does not have a cover protocol to mimic. Otherwise, our evaluation would simply investigate the similarity between our protocol and its cover protocol. Instead of measuring ScrambleSuit’s closeness to a mimicked protocol, we measure the deviation from its transported application, i.e., Tor. Intuitively, higher deviation would imply better obfuscation.

Philipp Winter, Tobias Pulls, and Juergen Fuss
ScrambleSuit: A Polymorphic Network Protocol to Circumvent Censorship (WPES13)
Tor Pluggable Transports

- **obfs4**
  - **Description:** Is a transport with the same features as **ScrambleSuit** but utilizing Dan Bernstein's **elligator2** technique for public key obfuscation, and the **rutor protocol** for one-way authentication. This results in a faster protocol.
  - **Language:** Go
  - **Maintainer:** Yawning Angel
  - **Evaluation:** obfs4 Evaluation

- **meek**
  - **Description:** Is a transport that uses HTTP for carrying bytes and TLS for obfuscation. Traffic is relayed through a third-party server (Google App Engine). It uses a trick to talk to the third party so that it looks like it is talking to an unblocked server.
  - **Language:** Go
  - **Maintainer:** David Fifield
  - **Evaluation:** meek Evaluation

- **Format-Transforming Encryption (FTE)**
  - **Description:** It transforms Tor traffic to arbitrary formats using their language descriptions. See the research paper.
  - **Language:** Python/C++
  - **Maintainer:** Kevin Dyer
  - **Evaluation:** FTE Evaluation

- **ScrambleSuit**
  - **Description:** Is a pluggable transport that protects against follow-up probing attacks and is also capable of changing its network fingerprint (packet length distribution, inter-arrival times, etc.).
  - **Language:** Python
  - **Maintainer:** Philipp Winter
  - **Evaluation:** ScrambleSuit Evaluation
Tor Pluggable Transports

- **obfs4**
  - **Description:** Is a transport with the same features as [ScrambleSuit](#) but utilizing Dan Bernstein's [elligator2](#) technique for public key obfuscation, and the [ntor protocol](#) for one-way authentication. This results in a faster protocol.
  - **Language:** Go

---

Undeployed PTs

These Pluggable Transports exist but are not deployed as part of the Tor Browser.

- **F**
  - **Description:** Look-like-nothing pluggable transport (in [obfsproxy](#))
  - **Language:** Python
  - **Notes:** Superseded by obfs3
  - **Maintainer:** as
  - **Evaluation:** [obfs3 Evaluation](#)

- **obfs2**
  - **Description:** Look-like-nothing pluggable transport (in [obfsproxy](#))
  - **Language:** Python
  - **Notes:** Superseded by obfs3
  - **Maintainer:** as
  - **Evaluation:** [obfs2 Evaluation](#)
  - **Maintainer:** Kevin Dyer
  - **Evaluation:** [FTE Evaluation](#)

- **ScrambleSuit**
  - **Description:** Is a pluggable transport that protects against follow-up probing attacks and is also capable of changing its network fingerprint (packet length distribution, inter-arrival times, etc.).
  - **Language:** Python
  - **Maintainer:** Philipp Winter
  - **Evaluation:** [ScrambleSuit Evaluation](#)
April 2016

- 77.1% vanilla
- 6.5% obf3+obf4+ssuit
- 6.3% obf3+fte+obf4+ssuit
- 4.4% obf3+fte+obf4+ssuit
- 3% obf3+obf4
- 1.6% obf3+ssuit
- 1.4% obf4
- 1.2% OTHER

Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)
April 2016

- 77.1% vanilla
- 6.5% obf3+obf4+ssuit
- 6.3% obf3+fte+obf4+ssuit
- 4.4% obf3+fte+obf4+ssuit
- 3% obf3+obf4
- 1.6% obf3+ssuit
- 1.4% obf4
- 1.2% OTHER

Blockable!
PT DEPLOYMENT

April 2016

77.1% vanilla
6.5% obf3+obf4+ssuit
6.3% obf3+fte+obf4+ssuit
4.4% obf3+fte+obf4+ssuit
3% obf3+obf4
1.6% obf3+ssuit
1.4% obf4
1.2% OTHER

Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)
<table>
<thead>
<tr>
<th>PT</th>
<th>Used Brid.</th>
<th>Clients</th>
<th>Top 20 (Default)</th>
<th>Total Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanilla</td>
<td>1,967</td>
<td>14,939</td>
<td>5.6% (0.0%) [0]</td>
<td>1.2% [21]</td>
</tr>
<tr>
<td>obfs2</td>
<td>13</td>
<td>158</td>
<td>100.0% (25.8%) [1]</td>
<td>25.8% [1]</td>
</tr>
<tr>
<td>obfs3</td>
<td>898</td>
<td>63,088</td>
<td>92.0% (90.8%) [4]</td>
<td>90.8% [4]</td>
</tr>
<tr>
<td>obfs4</td>
<td>792</td>
<td>204,095</td>
<td>95.4% (94.7%) [11]</td>
<td>94.7% [11]</td>
</tr>
<tr>
<td>ssuit</td>
<td>467</td>
<td>4,483</td>
<td>52.4% (46.3%) [1]</td>
<td>46.3% [1]</td>
</tr>
<tr>
<td>meek</td>
<td>4</td>
<td>22,685</td>
<td>100.0% (~100%) [3]</td>
<td>~100% [3]</td>
</tr>
</tbody>
</table>

**TABLE III.** BRIDGE IMPORTANCE PER PT (Apr’16).
### Usage of PTs – Ranking

<table>
<thead>
<tr>
<th>PT</th>
<th>Used Brid.</th>
<th>Clients</th>
<th>Top 20 (Default)</th>
<th>Total Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>vanilla</td>
<td>1,967</td>
<td>14,939</td>
<td>5.6% (0.0%) [0]</td>
<td>1.2% [21]</td>
</tr>
<tr>
<td>obfs2</td>
<td>13</td>
<td>158</td>
<td>100.0% (25.8%) [1]</td>
<td>25.8% [1]</td>
</tr>
<tr>
<td>obfs3</td>
<td>898</td>
<td>63,088</td>
<td>92.0% (90.8%) [4]</td>
<td>90.8% [4]</td>
</tr>
<tr>
<td>obfs4</td>
<td>792</td>
<td>204,095</td>
<td>95.4% (94.7%) [11]</td>
<td>94.7% [11]</td>
</tr>
<tr>
<td>ssuit</td>
<td>467</td>
<td>4,483</td>
<td>52.4% (46.3%) [1]</td>
<td>46.3% [1]</td>
</tr>
<tr>
<td>meek</td>
<td>4</td>
<td>22,685</td>
<td>100.0% (~100%) [3]</td>
<td>~100% [3]</td>
</tr>
</tbody>
</table>

**TABLE III. BRIDGE IMPORTANCE PER PT (APR’16).**

94% obs4 in default!

Useless reply protection...

---

Srdjan Matic, Carmela Troncoso, and Juan Caballero
Dissecting Tor Bridges: a Security Evaluation of their Private and Public Infrastructures (NDSS 2017)
Takeaways

• Privacy is not only about accuracy, False positives matter

• Semantic attacks may not work as well as thought

• Obfuscating is as hard as mimic
  • Too random is as noticeable as non random
  • ML to learn patterns is very powerful
IDENTIFYING THE FLOW: WEBSITE FINGERPRINTING

**Flow properties:**
- length, inter-arrival times, bursts, Obfuscation, mimic
IDENTIFYING THE FLOW: WEBSITE FINGERPRINTING

Flow properties:
- length, inter-arrival times, bursts,

Obfuscation, mimic, hide a CLASS
Flow properties: length, inter-arrival times, bursts, Obfuscation, mimic hide a CLASS

Identifying a particular FLOW
IDENTIFYING THE FLOW: WEBSITE FINGERPRINTING

Flow properties:
- length, inter-arrival times, bursts

Identifying a particular FLOW

Obfuscation, mimic
Hide a CLASS

Why does it work?
Flow properties:
- length, inter-arrival times, bursts,
- Obfuscation, mimic, hide a CLASS

Identifying a particular FLOW

Why does it work?

Next week

Peek-a-Boo, I Still See You:
Why Efficient Traffic Analysis Countermeasures Fail
Dyer, Coull, Ristenpart, and Shrimpton.