TLS and Privacy

Håkon Jacobsen
hakoja@item.ntnu.no

November 14, 2016

Transport Layer Security (TLS)

- The world's most used security protocol
- The "S" in HTTP**S**, FTP**S**, SMTP**S**, ...
- $>$ 50% of Chrome and Firefox page loads are over HTTPS[1]
- Protects communication between a client and server at the transport layer (end-to-end)

[1] https://security.googleblog.com/2016/11/heres-to-more-https-on-web.html
https://twitter.com/0xjosh/status/786971412959420424

Transport Layer Security (TLS)

- The world's most used security protocol
- The "S" in HTTP**S**, FTP**S**, SMTP**S**, ...
- $>$ 50% of Chrome and Firefox page loads are over HTTPS[1]
- Protects communication between a client and server at the transport layer (end-to-end)
- However, TLS is *not* a privacy protocol!

[1] https://security.googleblog.com/2016/11/heres-to-more-https-on-web.html
https://twitter.com/0xjosh/status/786971412959420424

Transport Layer Security (TLS)

- The world's most used security protocol
- The "S" in HTTP**S**, FTP**S**, SMTP**S**, ...
- $>$ 50% of Chrome and Firefox page loads are over HTTPS[1]
- Protects communication between a client and server at the transport layer (end-to-end)
- However, TLS is *not* a privacy protocol!
  In Phil's words: TLS is crypto for security

[1] https://security.googleblog.com/2016/11/heres-to-more-https-on-web.html
https://twitter.com/0xjosh/status/786971412959420424

TLS Handshake

Client		Server
ClientHello	→	
	←	ServerHello Certificate key_share (g^y) CertificateVerify CertificateRequest*
key_share (g^x) Certificate* CertificateVerify* Finished	→	
	←	Finished
Application Data	↔	Application Data

$\star$ – only sent when using client authentication

▶ – encrypted under the final traffic key $tk \leftarrow H(g^{xy})$

TLS Record Layer

- Actual bits sent on the wire
- Each record is *tagged* with a *content type*:
  - `Application`
  - `Handshake`
  - `Alert`
  - `ChangeCipherSpec`
- Key from handshake is used to encrypt data
- But the tags are *not* encrypted

TLS 1.3

- Currently under development/standardization by IETF
- Aimed at improving the security and efficiency of TLS 1.2
- Deprecates broken ciphersuites
- Mandates* forward secrecy
- 0-RTT data

TLS 1.3

- Currently under development/standardization by IETF
- Aimed at improving the security and efficiency of TLS 1.2
- Deprecates broken ciphersuites
- Mandates* forward secrecy
- 0-RTT data
- What about privacy?

TLS 1.3

- Currently under development/standardization by IETF
- Aimed at improving the security and efficiency of TLS 1.2
- Deprecates broken ciphersuites
- Mandates* forward secrecy
- 0-RTT data
- What about privacy?
  - TLS ($\leq$ 1.2) supports anonymous key exchange
  - TLS 1.3 does not – a problem?

TLS Working Group Charter

Main design goals

- Develop a mode that encrypts as much of the handshake as is possible to reduce the amount of observable data to both passive and active attackers.
- ...
- ...
- The WG will consider the privacy implications of TLS 1.3 and where possible (balancing with other requirements) will aim to make TLS 1.3 more privacy-friendly, e.g. via more consistent application traffic padding, more considered use of long term identifying values, etc.

Current RFC version (draft 18)

Appendix D.1 (Handshake protocol)

- *Protection of endpoint identities.* The server's identity (certificate) should be protected against passive attackers. The client's identity should be protected against both passive and active attackers.

Appendix D.2 (Record protocol)

- *Length concealment.* Given a record with a given external length, the attacker should not be able to determine the amount of the record that is content versus padding.

TLS 1.2 Handshake

Client		Server
ClientHello	⟶	
	⟵	ServerHello Certificate key_share (g^y) CertificateVerify CertificateRequest*
key_share (g^x) Certificate* CertificateVerify* Finished	⟶	
	⟵	Finished
Application Data	⟷	Application Data

$\star$ – only sent when using client authentication

▶ – encrypted under the final traffic key $tk \leftarrow H(g^{xy})$

TLS 1.3 Handshake

```
            Client                                   Server

  ClientHello
  key_share (g^x)        ------------------------>

                                               ServerHello
                                               key_share (g^y)
                                               Certificate
                                               CertificateVerify
                                               CertificateRequest*
                         <------------------------ Finished
Certificate*
CertificateVerify*
Finished                 ------------------------>

  Application Data       <----------------------> Application Data
```

⋆ – only sent when using client authentication

▶ (orange) – encrypted under intermediate *unauthenticated* key $hs \leftarrow H_1(g^{xy})$

▶ (green) – encrypted under the final *authenticated* traffic key $tk \leftarrow H_2(g^{xy})$

TLS 1.3 Record Layer

- Mostly similar to TLS 1.2 and below (including padding)
- But tags are now encrypted under the traffic key

Can TLS Provide Privacy?

Something always leaks...

- DNS-queries
- IP-addresses
- Packet lengths
- Bandwidth usage
- Total download (or upload) time

Can TLS Provide Privacy?

Something always leaks...

- DNS-queries
- IP-addresses
- Packet lengths
- Bandwidth usage
- Total download (or upload) time

Countermeasures

- Padding of plaintext
  - per packet random
  - per session random
  - all to MTU
  - other
- Send dummy data

Are Traffic Analysis Countermeasures Effective?

Website Fingerprinting in Onion Routing Based Anonymization Networks

Andriy Panchenko
Interdisciplinary Center for Security, Reliability and Trust
University of Luxembourg
http://lorre.uni.lu/~andriy/
{firstname.lastname}@uni.lu

Lukas Niessen
Computer Science dept.
RWTH Aachen University
lukas.niessen@rwth-aachen.de

Andreas Zinnen, Thomas Engel
Interdisciplinary Center for Security, Reliability and Trust
University of Luxembourg
{firstname.lastname}@uni.lu

ABSTRACT

Low-latency anonymization networks such as Tor and JAP claim to hide the recipient and the content of communications from a *local observer*, i.e., an entity that can eavesdrop the traffic between the user and the first anonymization node. Especially users in totalitarian regimes strongly depend on such networks to freely communicate. For these people, anonymity is particularly important and an analysis of the anonymization methods against various attacks is necessary to ensure adequate protection. In this paper we show that anonymity in Tor and JAP is not as strong as expected so far and cannot resist *website fingerprinting* attacks under certain circumstances. We first define features for website fingerprinting solely based on volume, time, and direction

General Terms

Security

Keywords

Anonymous Communication, Website Fingerprinting, Traffic Analysis, Pattern Recognition, Privacy

1. INTRODUCTION

Anonymous communication aims at hiding the relationship between communicating parties on the Internet. Thereby, anonymization is the technical basis for a significant number of users living in oppressive regimes [15] giving users the opportunity to communicate freely and, under certain circum-

Are Traffic Analysis Countermeasures Effective?

Website Fingerprinting in Onion Routing Based Anonymization Networks

Andriy Panchenko
Interdisciplinary Center for
Security, Reliability and Trust

Lukas Niessen
Computer Science dept.
RWTH Aachen University

Andreas Zinnen,
Thomas Engel
Interdisciplinary Center for

ABSTRACT

Low-latency ar... claim to hide t... tions from a l... drop the traffi... tion node. Esp... depend on such... people, anonym... of the anonymi... essay to ensur... that anonymity... so far and cann... certain circum... fingerprinting solely based on volume, time, and direction

... portunity to communicate freely and, under certain circum-

Page Set	True Positives	False Positives
Sexually explicit	56.0%	0.89%
Alexa top ranked	73.0%	0.05%
Alexa random	56.5%	0.23%

Table 1: True and false positive rate for Sexually explicit, Alexa top ranked and Alexa random of the Open-World Dataset

Are Traffic Analysis Countermeasures Effective?

Website Fingerprinting in Onion Routing Based Anonymization Networks

Peek-a-Boo, I Still See You:
Why Efficient Traffic Analysis Countermeasures Fail

Kevin P. Dyer*, Scott E. Coull†, Thomas Ristenpart‡, and Thomas Shrimpton*

*Department of Computer Science, Portland State University, Portland, USA. Email: {kdyer, teshrim}@cs.pdx.edu
† RedJack, LLC., Silver Spring, MD, USA Email: scott.coull@redjack.com
‡Department of Computer Sciences, University of Wisconsin-Madison, USA. Email: rist@cs.wisc.edu

***Abstract*—**
We consider the setting of HTTP traffic over encrypted tunnels, as used to conceal the identity of websites visited by a user. It is well known that traffic analysis (TA) attacks can accurately identify the website a user visits despite the use of encryption, and previous work has looked at specific attack/countermeasure pairings. We provide the first comprehensive analysis of general-purpose TA countermeasures. We show that nine known countermeasures are vulnerable to simple attacks that exploit coarse features of traffic (e.g., to-

manipulate whole streams of packets in order to precisely mimic the distribution of another website's packet lengths.

The seemingly widespread intuition behind these countermeasures is that they patch up the most dangerous side channel (packet lengths) and so provide good protection against TA attacks, including website identification. Existing literature might appear to support this intuition. For example, Liberatore and Levine [10] show that padding

Are Traffic Analysis Countermeasures Effective?

Website Fingerprinting in Onion Routing Based Anonymization Networks

Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail

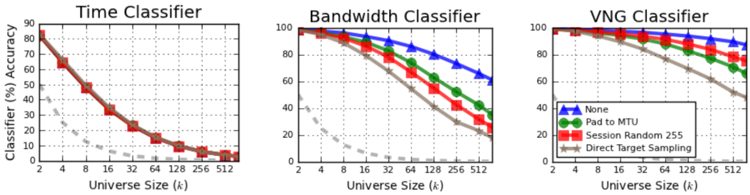

Figure 9. The average accuracy against the raw encrypted traffic (None), and the best countermeasures from each type, as established in Section V. (left) the time-only classifier. (middle) the bandwidth only classifier. (right) the VNG ("burstiness") classifier.

The End

Thank you!