

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2020

I. Symeonidis
University of Luxembourg
B. Hoeneisen
Ucom.ch
October 31, 2019

Privacy and Security Threat Analysis for Private Messaging
draft-symeonidis-pearg-private-messaging-threats-00

Abstract

Modern email and instant messaging applications offer private communications between users. As IM and Email network designs become more similar, both share common concerns about security and privacy of the information exchanged. However, the solutions available to mitigate these threats and to comply with the requirements may differ. The two communication methods are, in fact, built on differing assumptions and technologies. Assuming a scenario of untrusted servers, we analyze threats against message delivery and storage, the requirements that these systems need, and the solutions that exist in order to help implement secure and private messaging. From the discussed technological challenges and requirements, we aim to derive an open standard for private messaging.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	4
1.2. Terms	4
2. System Model	4
2.1. Entities	4
2.2. Assets and Functional Requirements	5
3. Threat Analyses and Requirements	5
3.1. Adversarial Model	5
3.2. Assumptions	6
3.3. Security Threats and Requirements	6
3.3.1. Spoofing and Entity Authentication	6
3.3.2. Information Disclosure and Confidentiality	7
3.3.3. Tampering With Data and Data Authentication	7
3.3.4. Repudiation and Accountability (Non-Repudiation)	7
3.3.5. Elevation of Privilege and Authorization	8
3.4. Privacy Threats and Requirements	8
3.4.1. Identifiability - Anonymity	8
3.4.2. Linkability - Unlinkability	8
3.4.3. Detectability and Observability - Undetectability	9
3.5. Information Disclosure - Confidentiality	9
3.6. Non-repudiation and Deniability	9
3.6.1. Policy Non-compliance and Policy compliance	10
4. Security Considerations	10
5. Privacy Considerations	10
6. Future Key Challenges	10
7. IANA Considerations	10
8. Acknowledgments	10
9. References	10
9.1. Normative References	10
9.2. Informative References	11
Appendix A. Document Changelog	12
Appendix B. Open Issues	12
Authors' Addresses	12

1. Introduction

Private messaging should ensure that, in an exchange of messages between (two) peers, no one but the sender and the receiver of the communication will be capable of reading the messages exchanged at any (current, future or past) time. Essentially, no one but the communicating peers should ever have access to the messages during transit such as Telecom, Internet providers, or intermediary parties, and storage such as messaging servers. As private messaging, we are referring to Instant Messaging (IM) [RFC2779], such as WhatsApp and Signal, and Emailing applications, such as the centralized Protonmail and the fully decentralized pEp [I-D.birk-pep].

The aim of this document is to provide an open standard for private messaging requirements, as well as a unified evaluation framework. The framework catalogues security and privacy threats and the corresponding, to threats, requirements. IM and Email applications have common feature design characteristics and support a common set of information assets for transmission during communication between peers. For example, applications for both systems should support message exchange of text and files (e.g., attachments) in a private messaging manner.

Despite having common characteristics, IM and Email have network design divergences in areas such as responsiveness and synchronicity. For example, low-latency and synchronous were the common features for instant messaging and high-latency and asynchronous for email. As IM and Email network designs become more similar, approaches to security and privacy should be able to address both types of communications. Current IM applications tend to be asynchronous, allowing delivery of messages when the communicating parties are not at the same time online.

Solutions available to implement private messaging in the two types of applications may call for different mitigation mechanisms and design choices. For instance, confidentiality can be preserved in multiple ways and with various cryptographic primitives. As design choices, it depends on the expected level of protection and the background of the user. For instance, for users whose lives may be at stake, such as journalists, whistleblowers, or political dissidents, the design choices for requirements and mitigation mechanisms can be (and often are) much more advanced than those for organizations and general end-users. Despite this distinction, privacy and security on the internet are Human Rights, and easily-enabled means to protect these rights need to exist. But in cases where stronger protections are required, usability may come second to more robust protection.

The objectives of this document are to create an open standard for secure messaging requirements. The open standard for private messaging aims to serve as a unified evaluation framework, including an adversarial model, threats, and requirements. With this document, we catalogue the threats and requirements for implementing secure and private messaging systems. In this current version, we discuss two key design features of IM and Email, message delivery and storage/archival. This draft is an ongoing work in progress, and the list of requirements discussed here are not exhaustive. However, our work already shows an emerging and rich set of security and privacy challenges.

Of course, IM additionally can support voice/video calls, which is an additional feature/asset under which a threat assessment and requirements can be evaluated.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2. Terms

The following terms are defined for the scope of this document:

- o Man-in-the-middle (MITM) attack: cf. [RFC4949], which states: "A form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data to masquerade as one or more of the entities involved in a communication association."

2. System Model

2.1. Entities

- o Users: The communicating parties who exchange messages, typically referred to as senders and receivers.
- o Messaging operators and network nodes: The communicating service providers and network nodes that are responsible for message delivery and synchronization.
- o Third parties: Any other entity who interacts with the messaging system.

2.2. Assets and Functional Requirements

This section outlines a private messaging system. It describes the functionalities that needs to support and the information that can be collected by the system as assets from users. We follow the requirements extracted from real world systems and applications as well as from the academic literature for email and instant messaging [Unger] [Ermoshina] [Clark].

Assets:

- o Content: text, files (e.g., attachments), voice/video
- o Identities: sender/receiver identity, contact list
- o Metadata: sender/receiver, timing, frequency, packet size

Functionalities:

- o [Email/IM] Messages: send and receive text + attachments
 - * Peer or group: more than 2 participants communicating
- o [IM] Voice / video call
- o [Email/IM] Archive and search: of messages and attachments
- o [Email/IM] Contacts: synchronisation and matching
- o [Email/IM] Multi-device support: synchronisation across multiple devices

3. Threat Analyses and Requirements

This section describes a set of possible threats. Note that typically not all threats can be addressed in a system, due to conflicting requirements.

3.1. Adversarial Model

An adversary is any entity who leverages threats against the communication system, whose goal is to gain improper access to the message content and users' information. They can be anyone who is involved in communication, such as users of the system, message operators, network nodes, or even third parties.

- o Internal - external: An adversary can seize control of entities within the system, such as extracting information from a specific

entity or preventing a message from being sent. An external adversary can only compromise the communication channels themselves, eavesdropping and tampering with messaging such as performing Man-in-the-Middle (MitM) attacks.

- o Local - global: A local adversary can control one entity that is part of a system, while a global adversary can seize control of several entities in a system. A global adversary can also monitor and control several parts of the network, granting them the ability to correlate network traffic, which is crucial in performing timing attacks.
- o Passive - active: A passive attacker can only eavesdrop and extract information, while an active attacker can tamper with the messages themselves, such as adding, removing, or even modifying them.

Attackers can combine these adversarial properties in a number of ways, increasing the effectiveness - and probable success - of their attacks. For instance, an external global passive attacker can monitor multiple channels of a system, while an internal local active adversary can tamper with the messages of a targeted messaging provider [Diaz].

3.2. Assumptions

In this current work, we assume that end points are secure such that the mobile devices of the users. Moreover, we assume that an adversary cannot break any of the underline cryptographic primitives.

3.3. Security Threats and Requirements

3.3.1. Spoofing and Entity Authentication

Spoofing occurs when an adversary gains improper access to the system upon successfully impersonating the profile of a valid user. The adversary may also attempt to send or receive messages on behalf of that user. The threat posed by an adversary's spoofing capabilities is typically based on the local control of one entity or a set of entities, with each compromised account typically is used to communicate with different end-users. In order to mitigate spoofing threats, it is essential to have entity authentication mechanisms in place that will verify that a user is the legitimate owner of a messaging service account. The entity authentication mechanisms typically rely on the information or physical traits that only the valid user should know/possess, such as passwords, valid public keys, or biometric data like fingerprints.

3.3.2. Information Disclosure and Confidentiality

An adversary aims to eavesdrop and disclose information about the content of a message. They can attempt to perform a man-in-the-middle attack (MitM). For example, an adversary can attempt to position themselves between two communicating parties, such as gaining access to the messaging server and remain undetectable while collecting information transmitted between the intended users. The threat posed by an adversary can be from local gaining control of one point of a communication channel such as an entity or a communication link within the network. The adversarial threat can also be broader in scope, such as seizing global control of several entities and communication links within the channel. That grants the adversary the ability to correlate and control traffic in order to execute timing attacks, even in the end-to-end communication systems [Tor]. Therefore, confidentiality of messages exchanged within a system should be guaranteed with the use of encryption schemes

3.3.3. Tampering With Data and Data Authentication

An adversary can also modify the information stored and exchanged between the communication entities in the system. For instance, an adversary may attempt to alter an email or an instant message by changing the content of them. As a result, it can be anyone but the users who are communicating, such as the message operators, the network node, or third parties. The threat posed by an adversary can be in gaining local control of an entity which can alter messages, usually resulting in a MitM attack on an encrypted channel. Therefore, no honest party should accept a message that was modified in transit. Data authentication of messages exchanged needs to be guaranteed, such as with the use of Message Authentication Code (MAC) and digital signatures.

3.3.4. Repudiation and Accountability (Non-Repudiation)

Adversaries can repudiate, or deny, the status of the message to users of the system. For instance, an adversary may attempt to provide inaccurate information about an action performed, such as about sending or receiving an email. An adversary can be anyone who is involved in communicating, such as the users of the system, the message operators, and the network nodes. To mitigate repudiation threats, accountability, and non-repudiation of actions performed must be guaranteed. Non-repudiation of action can include proof of origin, submission, delivery, and receipt between the intended users. Non-repudiation can be achieved with the use of cryptographic schemes such as digital signatures and audit trails such as timestamps.

3.3.5. Elevation of Privilege and Authorization

An adversary may attempt to elevate privileges aiming to gain access to the assets of other users or the resources of the system. For instance, an adversary may attempt to become an administrator of a message group or a superuser of the system aiming at retrieving users' messages or executing operations as a superuser. Therefore, authorization mechanisms such as access control lists that comply with the principle of least privilege for user accounts and processes should be applied.

3.4. Privacy Threats and Requirements

3.4.1. Identifiability - Anonymity

Identifiability is defined as the extent to which a specific user can be identified from a set of users, which is the identifiability set. Identification is the process of linking information to allow the inference of a particular user's identity [RFC6973]. An adversary can identify a specific user associated with Items of Interest (IOI), which include items such as the ID of a subject, a sent message, or an action performed. For instance, an adversary may identify the sender of a message by examining the headers of a message exchanged within a system. To mitigate identifiability threats, the anonymity of users must be guaranteed. Anonymity is defined from the attackers perspective as the "attacker cannot sufficiently identify the subject within a set of subjects, the anonymity set" [Pfitzmann]. Essentially, in order to make anonymity possible, there always needs to be a set of possible users such that for an adversary the communicating user is equally likely to be of any other user in the set [Diaz]. Thus, an adversary cannot identify who is the sender of a message. Anonymity can be achieved with the use of pseudonyms and cryptographic schemes such as anonymous remailers (i.e., mixnets), anonymous communications channels (e.g., Tor), and secret sharing.

3.4.2. Linkability - Unlinkability

Linkability occurs when an adversary can sufficiently distinguish within a given system that two or more IOIs such as subjects (i.e., users), objects (i.e., messages), or actions are related to each other [Pfitzmann]. For instance, an adversary may be able to relate pseudonyms by analyzing exchanged messages and deduce that the pseudonyms belong to one user (though the user may not necessarily be identified in this process). Therefore, unlinkability of IOIs should be guaranteed through the use of pseudonyms as well as cryptographic schemes such as anonymous credentials.

3.4.3. Detectability and Observability - Undetectability

Detectability occurs when an adversary is able to sufficiently distinguish an IOI, such as messages exchanged within the system, from random noise [Pfitzmann]. Observability occurs when that detectability occurs along with a loss of anonymity for the entities within that same system. An adversary can exploit these states in order to infer linkability and possibly identification of users within a system. Therefore, undetectability of IOIs should be guaranteed, which also ensures unobservability. Undetectability for an IOI is defined as that "the attacker cannot sufficiently distinguish whether it exists or not." [Pfitzmann]. Undetectability can be achieved through the use of cryptographic schemes such as mix-nets and obfuscation mechanisms such as the insertion of dummy traffic within a system.

3.5. Information Disclosure - Confidentiality

Information disclosure - or loss of confidentiality - about users, message content, metadata or other information is not only a security but also a privacy threat that a communicating system can face. For example, a successful MitM attack can yield metadata that can be used to determine with whom a specific user communicates with, and how frequently. To guarantee the confidentiality of messages and prevent information disclosure, security measures need to be guaranteed with the use of cryptographic schemes such as symmetric, asymmetric or homomorphic encryption and secret sharing.

3.6. Non-repudiation and Deniability

Non-repudiation can be a threat to a user's privacy for private messaging systems, in contrast to security. As discussed in section 6.1.4, non-repudiation should be guaranteed for users. However, non-repudiation carries a potential threat vector in itself when it is used against a user in certain instances. For example, whistle-blowers may find non-repudiation used against them by adversaries, particularly in countries with strict censorship policies and in cases where human lives are at stake. Adversaries in these situations may seek to use shreds of evidence collected within a communication system to prove to others that a whistle-blowing user was the originator of a specific message. Therefore, plausible deniability is essential for these users, to ensure that an adversary can neither confirm nor contradict that a specific user sent a particular message. Deniability can be guaranteed through the use of cryptographic protocols such as off-the-record messaging.

3.6.1. Policy Non-compliance and Policy compliance

Policy non-compliance can be a threat to the privacy of users in a private messaging system. An adversary, can attempt to process information about users unlawfully and not-compliant to regulations. It may attempt to collect and process information of users exchanged in emails without the users' notification and explicit consent. That can result in unauthorized processing of users information under the General Data Protection Regulation resulting in of such as profiling, advertisement and censorship. Therefore, data protection policy compliance must be guaranteed. It can be achieved with auditing such as with Data Protection Impact Assessment considering [GDPR].

4. Security Considerations

Relevant security considerations are outlined in Section 3.3.

5. Privacy Considerations

Relevant privacy considerations are outlined in Section 3.4.

6. Future Key Challenges

Reducing metadata leakage and standardization (i.e. prevent further fragmentation).

7. IANA Considerations

This document requests no action from IANA.

[[RFC Editor: This section may be removed before publication.]]

8. Acknowledgments

The authors would like to thank the following people who have provided feedback or significant contributions to the development of this document: Athena Schumacher, Claudio Luck, Hernani Marques, Kelly Bristol, Krista Bennett, and Nana Karlstetter.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<https://www.rfc-editor.org/info/rfc4949>>.

9.2. Informative References

- [Clark] Clark, J., van Oorschot, P., Ruoti, S., Seamons, K., and D. Zappala, "Securing Email", CoRR abs/1804.07706, 2018.
- [Diaz] Diaz, C., Seys, St., Claessens, J., and B. Preneel, "Towards Measuring Anonymity", PET Privacy Enhancing Technologies, Second International Workshop, San Francisco, CA, USA, April 14-15, 2002, Revised Papers, pp. 54-68, 2002.
- [Ermoshina] Ermoshina, K., Musiani, F., and H. Halpin, "End-to-End Encrypted Messaging Protocols: An Overview", INSCI 2016: pp. 244-254, 2016.
- [GDPR] "General Data Protection Regulation 2016/680 of the European Parliament and of the Council (GDPR).", Official Journal of the European Union, L 119/89, 4.5.2016 , April 2016, <<https://eur-lex.europa.eu/eli/dir/2016/680/oj>>.
- [I-D.birk-pep] Marques, H., Luck, C., and B. Hoeneisen, "pretty Easy privacy (pEp): Privacy by Default", draft-birk-pep-04 (work in progress), July 2019.
- [Pfitzmann] Pfitzmann, A. and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management", 2010, <<https://nyuscholars.nyu.edu/en/publications/sok-secure-messaging>>.
- [RFC2779] Day, M., Aggarwal, S., Mohr, G., and J. Vincent, "Instant Messaging / Presence Protocol Requirements", RFC 2779, DOI 10.17487/RFC2779, February 2000, <<https://www.rfc-editor.org/info/rfc2779>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.

- [Tor] Project, T., "One cell is enough to break Tor's anonymity", June 2019, <<https://blog.torproject.org/one-cell-enough-break-tors-anonymity/>>.
- [Unger] Unger, N., Dechand, S., Bonneau, J., Fahl, S., Perl, H., Goldberg, I., and M. Smith, "SoK: Secure Messaging", IEEE Proceedings - 2015 IEEE Symposium on Security and Privacy, SP 2015, pages 232-249, July 2015, <<https://nyuscholars.nyu.edu/en/publications/sok-secure-messaging>>.

Appendix A. Document Changelog

- [[RFC Editor: This section is to be removed before publication]]
- o draft-symeonidis-pearg-private-messaging-threats-00:
 - * Initial version
 - * this document partially replaces draft-symeonidis-medup-requirements-00

Appendix B. Open Issues

- [[RFC Editor: This section should be empty and is to be removed before publication]]
- o Add more text on Group Messaging requirements
 - o Decide on whether or not "enterprise requirement" will go to this document

Authors' Addresses

Iraklis Symeonidis
University of Luxembourg
29, avenue JF Kennedy
L-1855 Luxembourg
Luxembourg

Email: iraklis.symeonidis@uni.lu
URI: https://wwwen.uni.lu/snt/people/iraklis_symeonidis

Bernie Hoeneisen
Ucom Standards Track Solutions GmbH
CH-8046 Zuerich
Switzerland

Phone: +41 44 500 52 40

Email: bernie@ietf.hoeneisen.ch (bernhard.hoeneisen AT ucom.ch)

URI: <https://ucom.ch/>

pearg
Internet-Draft
Intended status: Informational
Expires: May 7, 2020

I. Goldberg
University of Waterloo
T. Wang
HK University of Science and Technology
C. Wood
Apple, Inc.
November 04, 2019

Network-Based Website Fingerprinting
draft-wood-pearg-website-fingerprinting-00

Abstract

The IETF is well on its way to protecting connection metadata with protocols such as DNS-over-TLS and DNS-over-HTTPS, and work-in-progress towards encrypting the TLS SNI. However, more work is needed to protect traffic metadata, especially in the context of web traffic. In this document, we survey Website Fingerprinting attacks, which are a class of attacks that use machine learning techniques to attack web privacy, and highlight metadata leaks used by said attacks. We also survey proposed mitigations for such leakage and discuss their applicability to IETF protocols such as TLS, QUIC, and HTTP. We endeavor to show that Website Fingerprinting attacks are a serious problem that affect all Internet users, and we pose open problems and directions for future research in this area.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 7, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Background	3
3. Website Fingerprinting	4
4. Attacks	5
5. Base Rate Fallacy	8
6. Defenses	9
7. Open Problems and Directions	12
8. Protocol Design Considerations	14
9. Security Considerations	14
10. IANA Considerations	14
11. Informative References	14
Appendix A. Acknowledgements	20
Authors' Addresses	20

1. Introduction

Internet protocols such as TLS 1.3 [RFC8446] and QUIC [I-D.ietf-quic-transport] bring substantial improvements to end-users. The IETF engineered these with security and privacy in mind by encrypting more protocol messages using modern cryptographic primitives and algorithms, and engineering against flaws found in previous protocols, yielding several desirable security properties, including: forward-secure session key secrecy, downgrade protection, key compromise impersonation resistance, and protection of endpoint identities. Combined, these two protocols are set to protect a significant amount of Internet data. However, significant metadata leaks still exist for users of these protocols. Examples include plaintext TLS SNI and application-specific extensions (ALPN), as well as DNS queries. This information can be used by a passive attacker to learn information about the contents of an otherwise encrypted network connection. Recently, such information has also been studied

as a means of building unique user profiles [li2018can]. It has also been used to build flow classifiers that aid network management [foremski2014dns].

In the context of Tor, a popular low-latency anonymity network, a common class of attacks that use metadata for such inference is called Website Fingerprinting (WF). These attacks use machine learning techniques built with features extracted from metadata such as traffic patterns to attack web (browsing) privacy. Miller et al. [miller2014know] show how these attacks can be applied to web browsing traffic protected with HTTPS to reveal private information about users. Pironti et al. [pironti2012identifying] use similar attacks based on data sizes to identify individual social media clients using encrypted connections. Fingerprinting attacks using encrypted traffic analysis are also applicable to encrypted media streams, such as Netflix videos. (See work from Reed et al. [reed2017identifying] and Schuster et al. [schuster2017beauty] for examples of these attacks.) WF attacks have also been applied to other IETF protocols such as encrypted DNS, including dnscrypt, DNS-over-TLS, and DNS-over-HTTPS [siby2018dns][shulman2014pretty]. In the past, they have also been conducted remotely [gong2010fingerprinting], using buffer-based side channels in a victim's home router.

Protocols such as DNS-over-TLS and DNS-over-HTTPS [RFC8484], and work-in-progress towards encrypting the TLS SNI extension [I-D.ietf-tls-esni], help minimize metadata sent in cleartext on the wire. However, regardless of protocol and even network-layer fingerprinting mitigations, application layer specifics, e.g., web page sizes and client request patterns, reveal a noticeable amount of information to attackers. We argue that much more work is needed to protect encrypted connection metadata, especially in the context of web traffic.

In this document, we describe WF attacks in the context of IETF protocols such as TLS and QUIC. We survey WF attacks and highlight metadata features and classification techniques used to conduct said attacks. We also describe proposed mitigations for these attacks and discuss their applicability to IETF protocols. We conclude with a discussion of open problems and directions for future research and advocate for more work in this area.

2. Background

In this section we review how most secure Internet connections are made today. We omit custom configurations such as those using VPNs and proxies since they do not represent the common case for most Internet users. The following steps briefly describe the sequence of

events that normally occur when a web client, e.g., browser, curl, etc., connects to a website and obtains some resource. First an unencrypted DNS query is sent to an untrusted DNS recursive resolver to resolve a name to an IP address. Upon receipt, clients then open a TCP and TLS connection to the destination address. During this stage, metadata such as the TLS SNI and ALPN values are sent in cleartext. The SNI is used to denote the destination application or endpoint to which clients want to connect. Servers use this for several purposes, including selecting an appropriate certificate (one with the SNI name in the SubjectAlternativeName list) or routing to a different backend terminator. ALPN values are used to negotiate which application-layer protocol will be used on top of the TLS connection. Common values include "http/1.1", "h2", and (soon) "h3". Upon connection, clients then send HTTP messages to obtain the desired resource.

Connections look different (on the wire) with TLS 1.3, encrypted DNS via DNS-over-TLS or DNS-over-HTTPS, and encrypted SNI. DNS queries are encrypted to a (trusted) recursive resolver and TLS metadata such as SNI are encrypted in transit to the terminator. Despite the reduction in cleartext metadata sent over the wire, there still remains several sources of information that an adversary may use for malicious purposes, including: size and timing of DNS queries and responses, size and timing or application traffic, and connection attempts induced while loading a web resource, e.g., Javascript files. So while technologies such as Encrypted SNI, DoT, and DoH help protect some metadata, they are not complete solutions to the larger problem. In the following section, we discuss this overarching problem in detail.

3. Website Fingerprinting

Website Fingerprinting (WF) is a class of attacks that exploit metadata leakage to attack end-user privacy on the Internet. In the WF threat model, Adv is assumed to be a passive and local attacker. Local means that Adv can associate traffic with a given client. Examples include proxies to which clients directly connect. Passive means that Adv can only view traffic in transit. It cannot add, drop, or otherwise modify packets between the victim client and server(s). Use of reliable and encrypted transport protocols such as TLS limit on-path attackers to eavesdropping on encrypted packets. (In QUIC, however, reordering packets is possible.)

Traffic features used for classification include properties such as packet size, timing, direction, interarrival times, and burstiness, among many others [wang2016website]. Normally, features are restricted to those which are extractable as a passive eavesdropper, and not those which are viewable by modifying client or server

behavior. Specifically, this means that attacks such as CRIME [CRIME] and TIME [TIME], which rely on an attacker abusing TLS-layer compression to leak contents of an encrypted connection, are out of scope.

Website Fingerprinting attacks have evolved over the years through three phases: (1) Closed-world WF on SSL/TLS, (2) Closed-world WF on Tor, and (3) Open-world WF on Tor.

1. In the closed-world model, clients are assumed to only visit a small set of pages monitored by Adv. This is less realistic but easier to analyze than the open-world model discussed below, and so the earliest results achieved success on SSL/TLS in this model. (For a realistic attack, Adv would need to monitor every possible page of interest to each client, which is impractical.) Attacks against proxy-based privacy technologies such as VPNs and SSH tunneling, which has almost no effect on the network, falls under this category as well.
 2. Tor, an anonymity network built on onion routing, is harder to attack than SSL for several reasons; successful results on Tor thus came later. First, Tor pads all cells (Tor's application-layer datagrams) to the same constant size, removing unique packet lengths as a powerful feature for the attacker. Second, Tor imposes random network conditions upon the client due to random selection of proxies, so packet sequences are less likely to be consistent.
 3. In the open-world model, Adv wishes to learn whenever a victim client visits one of a select number of monitored pages [wang2016website]. Adversaries train classifiers in this model using monitored and non-monitored websites of their choosing. By definition, Adv cannot train using client-chosen pages. Clients then visit pages at will and Adv attempts to learn whenever a monitored page is visited, if any are at all. This is a realistic model capturing the fact that the set of pages any attacker would be interested in must necessarily be a small subset of the set of all pages. As this is a harder model to attack, successful results on this model came later.
4. Attacks
1. Closed-world WF on TLS: WF attacks date back to applications on SSL first inspired by Wagner and Schneier [wagner1996analysis], in which the authors observed that packet lengths reveal information about the underlying data. Subsequent attacks carried out by Cheng et al. [cheng1998traffic], Sun et al. [sun2002statistical], and Hintz [hintz2002fingerprinting]

continued to show access. These attacks assume Adv has knowledge of the target resource length(s), which is not always possible with techniques such as padding.

Bissias et al. [bissias2005privacy] use cross correlation of inter-packet times in one second time windows as an WF attack. Danezis [danezis2009traffic] model websites using a Hidden Markov Model (HMM) and use it, along with TLS traffic traces revealing only approximate lengths, to identify requested resources on a page. Their results vary the amount of information available to an adversary when building the HMM. Even in cases where resource popularity is omitted, which reflects the case where an adversary scrapes static websites, resource recall was high (86%). Liberatore and Levine [liberatore2006inferring] proposed two WF attacks using the Jaccard coefficient and the Naive Bayes classifier. Herrmann et al. [herrmann2009website] extended the work of Liberatore and Levine with a multinomial Naive Bayes classifier computed using three input frequency transformations. Results yielded higher accuracy than that of Liberatore and Levine. Herrmann's attack is the best in this category, but the authors assume packets which do not fill a MTU represent packet trailers. Therefore, uniqueness is only accurate modulo the MTU. Efficacy is limited if endpoints pad packets to the MTU or another fixed length. Modern protocols such as HTTP/2, QUIC, and TLS 1.3 all provide some form of application-controlled padding. (Note: These attacks are not successful on Tor.)

1. Closed-world WF on Tor: Shmatikov and Wang [shmatikov2006timing] presented a WF attack that exploits cross correlation of arrival packet counts in one second time windows. Lu et al. [lu2010website] developed a classifier based on the Levenshtein distance between ingress and egress packet lengths extracted from packet sequences. Distance is computed between strings of ingress and egress packet lengths. The training packet sequence with the closest distance to the testing packet sequence is deemed the match. Dyer et al. [dyer2012peek] used a Naive Bayes classifier trained with a reduced set of features, including total response transmission time, length of packets (in each direction), and burst lengths. (Wang [wang2016website] notes that measuring burst lengths in Tor is difficult given the presence of SENDME cells for flow control.) This approach did not yield any measurable improvements over the SVM classifier from Panchenko et al. Cai et al. [cai2012touching] extend the work of Lu et al. by adding transpositions to the Levenshtein distance computation and normalizing the result, yielding what the authors refer to as the Optimal String Alignment Distance (OSAD). Before feature extraction, the authors round TCP packet lengths to the nearest multiple of 600B as an estimate of the number of Tor cells.

Wang et al. [wang2013improved] tuned the OSAD-based attack to improve its accuracy. Specific changes include use of Tor cells instead of TCP packets for packet and burst lengths, as well as heuristics to remove SENDME cells (those not carrying application data) from flows to recover true burst lengths. The authors also modified the distance computation by removing substitutions, increasing the weight for egress packets, and varying the transposition cost across the packet sequence (large weights at the beginning of a trace, and smaller weights near the end, where variations are expected across repeated page loads.) Wang et al. also developed an alternate classifier with lower accuracy yet superior performance (quadratic to linear time complexity). It works by minimizing the sum of two costs: sequence transpositions and sequence deletions or insertions. These two costs are computed separately, in contrast to the first approach which computes them simultaneously.

Hayes et al. [hayes2016k] developed an attack called k-fingerprinting, which uses a k-NN classifier with features ranked by random decision forests. Their feature set includes timing information, e.g., statistics on packets per second, among the higher ranked features. (Higher ranked features have more weight in the classification phase.) Yan et al. [yan2018feature] used similar (manually curated) features with a CNN-based classifier. Time-based features were among the more effective features identified. Rahman et al. [rahman2019tik] improved time-based features by focusing on bursts, e.g., burst length, variance, inter-burst delay, etc., rather than more granular per-packet statistics. (The latter tend to vary for inconsistencies across packet traces for websites.) This improved accuracy of existing Deep Learning attacks from Sirinam et al. [sirinam2018deep], especially when coupled with packet direction information.

1. Open-world WF on Tor and TLS: Panchenko et al. [panchenko2011website] were the first to use a support vector machine (SVM) classifier trained with web domain-specific features, such as HTML document sizes, as well as packet lengths. Wang et al. [wang2014effective] also developed an attack using a k-Nearest Neighbors (k-NN) classifier, which is a supervised machine learning algorithm, targeting the open world setting. The classifier extracts a large number of features from packet sequences, including raw (ingress and egress) packet counts, unique packet lengths, direction, burst lengths, and inter-packet times, among others. (There are 4226 features in total.) The k-NN distance metric is computed as the sum of weighted feature differences.

Kota et al. [abe2016fingerprinting] were the first to use Deep Learning (DL) methods based on Stacked Denoising Autoencoders for WF

attacks. (Autoencoders reduce feature input dimensions when stacked.) Kota et al. form input vectors from Tor cell directions (+1 or -1). They use no other features. Using a (small) data set from Wang [wang2016website], the classifier achieves a 86% true positive rate and 2% false positive rate in the open world model. Rimmer et al. [rimmer2018automated] applied DL for automated feature generation and classifier construction. Trained with 2,500 traces per website, their system achieves 96.3% accuracy in the open world model. Recently, Bhat et al. [bhat2018var], Oh et al. [oh2017pfp], and Sirinam et al. [sirinam2018deep] used Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs) for WF attacks. Results from Sirinam et al. show the best results - 98% on Tor without recent defenses (in Section {{defenses}}) - while performing favorably when select defenses are used for both open and closed world models.

Yan et al. [yan2018feature] studied manual high-information feature extraction from packet traces. They "exhaustively" examined different levels of features, including packet, burst, TCP, port, and IP address, summing to 35,683 in total, and distilled them into a diverse set of uncorrelated features for eight different communication scenarios. Rahman [rahman2018using] studied the utility of features derived from packet interarrival times, including: median interarrival time (per burst), burst packet arrival time variance, cross-burst interarrival median differences, and others. Using a CNN, results show that these features yield a non-negligible increase in WF attack accuracy.

5. Base Rate Fallacy

For all WF attacks, one limitation worth highlighting is the base rate fallacy. This can be summarized as follows: highly accurate classifiers with a reliable false positive rate (FPR) decrease in efficacy as the world size increases. Juarez et al. [juarez2014critical] studied its impact by measuring the Bayesian detection rate (BDR) in comparison to the FPR as a function of world size. As the world size increases, the BDR approaches 0 while the FPR remains stable, meaning that the probability of incorrect classifier results increase as well. Juarez et al. partially address the base rate fallacy problem by adding a confirmation step to their classifier. Another problem is that web content is (increasingly) dynamic. Most WF attacks, especially those in closed world models, assume that traces are static. However, Juarez et al. [juarez2014critical] show this is not the case even for "simple" pages such as google.com. Thus, due to the base fallacy rate and dynamic nature of content, classifiers require continual retraining in order to ensure accuracy.

6. Defenses

WF defenses are deterministic or randomized algorithms that take as input application data or packet sequences and return modified application data or packet sequences. Viable defenses seek to minimize the transformation cost and maximum (theoretical and perfect) attacker accuracy. Naïve defenses such as sending a constant stream of (possibly random) bytes between client and server may be effective though clearly not viable from a cost perspective. Relevant cost metrics include bandwidth overhead, added time or latency (and its impact on related metrics such as page load time), and even CPU cost, though the latter is often ignored in favor of the former two. Wang [wang2016website] describe defenses as either limited or general. A limited defense is one which only helps mitigate specific WF attacks by transforming packets in a way to obviate a particular (set of) feature(s) used by said attacks. In contrast, general defenses help mitigate a variety of attacks.

Several general defenses have been proposed, including BuFLO [dyer2012peek], which pads packets to a fixed length of 1500B (the normal MTU) and schedules packets for transmission at fixed period intervals (and sends fake data if nothing is yet available). Tamaraw [wang2014comparing] is an improvement over BuFLO that uses two different fixed lengths for packet transmission, rather than one, to save on bandwidth overhead. Tamaraw also uses two different scheduling rates for ingress and egress packets. The authors chose to make the ingress packet period smaller than the egress packet period since HTTP responses are often larger in size and count – if HTTP Push is used – than requests. While provably correct, both BuFLO and Tamaraw limit the rate at which clients send traffic, and requires all clients to send at a uniform rate. Both requirements therefore make it difficult to apply as a generic defense in IETF protocols.

Wang et al. also developed Supersequence [wang2016website], which attempts to approximate a bandwidth-optimal deterministic defense. This is done by casting the padding and flow control problem as the shortest common subsequence (SCS) of the transformed packet trace. Supersequence approximates the solution by learning the optimal packet scheduling rate; it uses the same padding scheme as Tamaraw.

Walkie-Talkie [wang2015walkie] is a collection of mechanisms for WF defense. It includes running the client (browser) in half-duplex mode to batch requests and responses together, as well as randomly padding traffic so as to mimic traffic of benign websites. It assumes knowledge of traffic patterns for benign websites, which can be information learned over time or provided by a cooperating peer. Goldberg and Wang also propose a "randomized" variant that pads real

bursts of requests and generates random request bursts according to a uniform distribution. The half-duplex mode could be implemented as an extension to a protocol such as HTTP/2, QUIC, or TLS.

Many limited defenses have also been proposed. We list prominent works below.

- o Shmatikov and Wang [shmatikov2006timing] developed adaptive padding which adds packets to mask inter-packet times. (This mechanism does not ever delay application data being sent, in contrast to other padding mechanisms such as BuFLO; see below.) Juarez et al. [juarez2015wtf][juarez2016toward] also created a WF defense based on adaptive padding called WTF-PAD. This variant uses application data and "gap" distribution to generate padding for delays. Specifically, when not sending application data, senders use the gap distribution to drive fake packet transmission. WTF-PAD can be run by a single endpoint, though it is assumed that both client and server participate. As mentioned above, protocols such as HTTP/2, QUIC, and TLS 1.3 offer a mechanism by which applications can send padding. WTF-PAD could therefore be implemented as an extension to any of these protocols, either by applications supplying padding distributions or the system learning them over time.
- o In the context of HTTP, Danezis [danezis2009traffic] proposed padding: URLs, content, and even HTML document structures to mask application data lengths.
- o Wright et al. [wright2009traffic] developed traffic morphing, which pads packets in such a way so as to make the sequence from one page have characteristics of another (non-monitored or benign) page. This technique requires application-specific knowledge about benign pages and is therefore best implemented outside of the transport layer.
- o Nithyanand et al. [nithyanand2014glove] developed a mechanism called Glove, wherein traces were first clustered and then morphed (via dummy insertion, packet merging, splitting, and delaying) to look indistinguishable within clusters. When used to protect the Alexa top 500 domains, Glove performs well with respect to bandwidth overhead when compared to BuFLO and CS-BuFLO. Varying the cluster size can tune Glove's bandwidth overhead.
- o Pironti et al. [pironti2012identifying] developed a TLS-based fragmentation and padding scheme designed to hide the length of application data within a certain range with record padding. The mechanism works by iteratively splitting application data into

variable sized segments. Applications can guide the range of viable lengths provided such information is available.

- o Luo et al. [luo2011https] created HTTPS with Obfuscation (HTTPOS), which is a client-side mechanism for obfuscating HTTP traffic. It uses the HTTP Range method to receive resources in chunks, TCP MSS to limit the size of individual chunks, and advertised window size to control the flow of chunks in transmission.
- o Panchenko et al. [panchenko2011website] developed Decoy, which is a simple mechanism that loads a benign page alongside a real page. This seeks to mask the real page load by properties of the "decoy" page. As with morphing, this defense requires application-specific knowledge about benign pages and is best implemented outside of the transport layer.
- o The Tor project implemented HTTP pipelining [perry2011experimental], which bundles egress HTTP/1.1 requests into batches of varying sizes with random orders. Batching requests to mask request and response sizes could be made easier with HTTP/2 [RFC7540], HTTP/3, and QUIC, since these protocol naturally support multiplexing. However, pipelining and batching may necessarily introduce latency delays that negatively impact the user experience.
- o Cherubin et al. [cherubin2017website] design two application-layer defenses called Application Layer Padding Concerns Adversaries (ALPaCA) and Lightweight application-Layer Masquerading Add-on (LLaMA). ALPaCA is a server-side defense that pads first-party content (deterministically or probabilistically) according to a known distribution. (Deterministic padding similar to Tamaraw performs worse than probabilistic padding.) LLaMA is similar to randomized pipelining, yet differs in that requests are also delayed (if necessary) and spurious requests are generated according to some probability distribution. Comparatively, ALPaCA yields a greater reduction in WF attack accuracy than LLaMA.
- o Lu et al. [lu2018dynaflow] designed DynaFlow, which is a defense that dynamically adjusts traffic flows using a combination of burst pattern morphing, constant traffic flow with flexible intervals, and burst padding. DynaFlow overhead is 40% less than that of Tamaraw and was shown to have similar benefits.
- o Rahman [rahman18gan] uses generative adversarial networks (GANs) to modify candidate burst properties of packet traces, i.e., by inserting dummy packets, such that they appear indistinguishable from other traces. Normally, the generator component in a GAN

uses random noise to produce information that matches a target data distribution as classified by the discriminator component. Rahman uses a modified GAN architecture wherein the generator produces padding (dummy packets) for input data such that the discriminator cannot distinguish it from noise, or a desired burst packet sequence. Preliminary results with the GAN trained and tested on defended traffic, i.e., traffic already subject to some form of WF defense, show a 9% increase in bandwidth and 15% decrease in attack accuracy (from 98% to 85% in a closed world setting).

- o Imani et al. [imanimockingbird] developed Mockingbird, a defense built on using generated adversarial examples, i.e., dummy traffic designed to disrupt classifier behavior, aimed towards model misclassification. When run on classifiers trained without adversarial examples, Mockingbird reduced state-of-the-art DF attacks and CUMUL attacks from [panchenko2016website] from 98% to 3% and 92% to 31%, respectively. Conversely, classifiers trained and hardened with adversarial examples only reduce attack accuracy from 95% to between 25-57%, respectively. Classification results for half-duplex traces, i.e., those in which traffic flows in half-duplex mode, are lower. Mockingbird's bandwidth overhead is tunable based on parameters that control the internal traffic shaping algorithm.

7. Open Problems and Directions

To date, WF attacks target clients running over Tor or some other anonymizing service, meaning that WF attacks are likely more accurate on normal TLS-protected connections. Moreover, attacks normally assume clients use HTTP/1.1 with parallel connections for parallel resource fetches. In recent years, however, protocols such as SPDY, HTTP/2, and QUIC with built-in padding support and multiplexed stream-based connections should make existing attacks more difficult to carry out. That said, it is unclear how exactly these protocol design trends will impact WF attacks. A non-exhaustive list of questions that warrant further research are below:

1. How does connection coalescing and consolidation affect WF attacks? Technologies such as DNS-over-HTTPS and ESNI favor architectures wherein a single network or connection can serve multiple origins or resources. With connection coalescing, traffic for multiple resources is sent on the same connection, thereby adding effects similar to that of the Decoy defense mechanism described in Section 6
2. To what extent does protocol multiplexing increase WF attack difficulty? Using a single connection with multiple streams to

avoid HoL blocking saves on connection startup and bandwidth costs while simultaneously mixing information from multiple requests and resources on the same connection.

3. How can protocol features such as HTTP Push be used to improve WF defense efficacy? Defenses without cooperative peer support often induce suboptimal bandwidth or latency costs. If both endpoints of a connection participate in the defense, even proactively with Push, perhaps this could be improved.
4. Can connection bootstrapping techniques such as those used by ESNI be used to distribute WF defense information? One possible approach is to distribute client padding profiles derived from CDN knowledge of serviced resources.
5. How can clients build, use, and possibly share WF defense information to benefit others?
6. How can applications package websites and subresources in such a way that limits unique information? For example, websites link to third party resources in an ad-hoc fashion, causing the subsequent trace of browser fetches to possibly uniquely identify the website.

Research into the above questions will help the IETF community better understand the extent to which WF attacks are a problem for Internet users in general.

It is worth mentioning that traffic-based WF attacks may not be required to achieve the desired goal of learning a connection's destination. Network connections by nature reveal information about endpoint behavior. The relationship between network address and domains, especially when stable and unique, are a strong signal for website fingerprinting. Trevisan et al. [trevisan2016towards] explored use of this signal as a reliable mechanism for website fingerprinting. They find that most major services (domains) have clearly associated IP address(es), though these addresses may change over time. Jiang et al. [jiang2007lightweight] and Tammaro et al. [tammaro2012exploiting] also previously came to the same conclusion. Address-based website fingerprinting was also explored by Patil and Borisov [patil2019can], wherein they showed that addresses, especially when grouped together as part of a single web page load, leak a substantial amount of information about the corresponding domain. Thus, in general, classifiers that rely solely on network addresses may be used to aid website fingerprinting attacks.

8. Protocol Design Considerations

New protocols such as TLS 1.3 and QUIC are designed with privacy-protections in mind. TLS 1.3, for example, has support for record-layer padding [RFC8446], albeit it is not used widely in practice. Despite this, TLS connections still leak metadata, including negotiated ciphersuites. (See [fordTLSMetadata] for a discussion of this issue.) QUIC is more aggressive in its use of encryption as both a mitigation for middlebox ossification and privacy enhancement. Future protocols should follow these trends when possible to remove unnecessary metadata from the network.

9. Security Considerations

This document surveys security and privacy attacks and defenses on encrypted TLS connections. It does not introduce, specify, or recommend any particular mitigation to the aforementioned attacks.

10. IANA Considerations

This document makes no IANA requests.

11. Informative References

- [abe2016fingerprinting]
"Fingerprinting attack on tor anonymity using deep learning", Asia-Pacific Advanced Network, 2016 , n.d..
- [backes2013preventing]
"Preventing Side-Channel Leaks in Web Traffic -- A Formal Approach", NDSS, 2013 , n.d..
- [bhat2018var]
"Var-CNN and DynaFlow -- Improved Attacks and Defenses for Website Fingerprinting", arXiv preprint arXiv:1802.10215 , n.d..
- [bissias2005privacy]
"Privacy vulnerabilities in encrypted HTTP streams", International Workshop on Privacy Enhancing Technologies, 2005 , n.d..
- [cai2012touching]
"Touching from a distance -- Website fingerprinting attacks and defenses", ACM conference on Computer and communications security, 2012 , n.d..

- [cheng1998traffic]
"Traffic analysis of SSL encrypted web browsing", n.d..
- [cherubin2017website]
"Website fingerprinting defenses at the application layer", Privacy Enhancing Technologies, 2017 , n.d..
- [coull2007web]
"On Web Browsing Privacy in Anonymized NetFlows", USENIX Security Symposium , n.d..
- [CRIME]
"The CRIME Attack", n.d.,
<https://www.ekoparty.org/archive/2012/CRIME_ekoparty2012.pdf>.
- [danezis2009traffic]
"Traffic Analysis of the HTTP Protocol over TLS", 2009 , n.d..
- [dyer2012peek]
"Peek-a-boo, i still see you -- Why efficient traffic analysis countermeasures fail", IEEE Symposium on Security and Privacy, 2012 , n.d..
- [fordTLSMetadata]
"Metadata Protection Considerations for TLS Present and Future", n.d., <<http://bford.info/pub/net/tlsmeta.pdf>>.
- [foremski2014dns]
"DNS-Class -- immediate classification of IP flows using DNS", International Journal of Network Management , n.d..
- [gong2010fingerprinting]
"Fingerprinting websites using remote traffic analysis", Proceedings of the 17th ACM conference on Computer and communications security , n.d..
- [hayes2016k]
"k-fingerprinting -- A Robust Scalable Website Fingerprinting Technique", USENIX Security Symposium, 2016 , n.d..
- [herrmann2009website]
"Website fingerprinting -- attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier", ACM workshop on Cloud computing security, 2009 , n.d..

- [hintz2002fingerprinting]
"Fingerprinting websites using traffic analysis",
International Workshop on Privacy Enhancing Technologies,
2002 , n.d..
- [I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed
and Secure Transport", draft-ietf-quic-transport-23 (work
in progress), September 2019.
- [I-D.ietf-tls-esni]
Rescorla, E., Oku, K., Sullivan, N., and C. Wood,
"Encrypted Server Name Indication for TLS 1.3", draft-
ietf-tls-esni-04 (work in progress), July 2019.
- [imanimockingbird]
"Mockingbird -- Defending Against Deep-Learning-Based
Website Fingerprinting Attacks with Adversarial Traces",
n.d., <<https://arxiv.org/pdf/1902.06626.pdf>>.
- [jiang2007lightweight]
"Lightweight application classification for network
management", SIGCOMM workshop on Internet network
management, 2007 , n.d..
- [juarez2014critical]
"A critical evaluation of website fingerprinting attacks",
ACM SIGSAC Conference on Computer and Communications
Security, 2014 , n.d..
- [juarez2015wtf]
"WTF-PAD -- toward an efficient website fingerprinting
defense for tor", CoRR, abs/1512.00524 , n.d., <<https://pdfs.semanticscholar.org/0f54/4d0845cb9f317722759dc49e1493ef30d83d.pdf>>.
- [juarez2016toward]
"Toward an efficient website fingerprinting defense",
European Symposium on Research in Computer Security,
2016 , n.d..
- [li2018can]
"Can We Learn What People Are Doing from Raw DNS
Queries?", IEEE INFOCOM 2018-IEEE Conference on Computer
Communications , n.d..

[liberatore2006inferring]

"Inferring the source of encrypted HTTP connections", ACM Conference on Computer and Communications Security, 2006 , n.d..

[lu2010website]

"Website fingerprinting and identification using ordered feature sequences", European Symposium on Research in Computer Security, 2010 , n.d..

[lu2018dynaflow]

"DynaFlow -- An Efficient Website Fingerprinting Defense Based on Dynamically-Adjusting Flows", Workshop on Privacy in the Electronic Society, 2018 , n.d..

[luo2011httpos]

"HTTPOS -- Sealing Information Leaks with Browser-side Obfuscation of Encrypted Flows", NDSS, 2011 , n.d..

[miller2014know]

"I know why you went to the clinic -- Risks and realization of https traffic analysis", International Symposium on Privacy Enhancing Technologies Symposium, 2014 , n.d..

[nithyanand2014glove]

"Glove -- A bespoke website fingerprinting defense", Proceedings of the 13th Workshop on Privacy in the Electronic Society , n.d..

[oh2017pfp]

"p-FP -- Extraction, Classification, and Prediction of Website Fingerprints with Deep Learning", n.d..

[panchenko2011website]

"Website fingerprinting in onion routing based anonymization networks", ACM workshop on Privacy in the electronic society, 2011 , n.d..

[panchenko2016website]

"Website Fingerprinting at Internet Scale", n.d.,
<<https://www.freehaven.net/anonbib/cache/fingerprinting-ndss2016.pdf>>.

[patil2019can]

"What can you learn from an IP?", n.d.,
<<https://irtf.org/anrw/2019/anrw2019-final44-acmpaginated.pdf>>.

- [perry2011experimental]
"Experimental defense for website traffic fingerprinting",
n.d., <[https://blog.torproject.org/
experimental-defense-website-traffic-fingerprinting](https://blog.torproject.org/experimental-defense-website-traffic-fingerprinting)>.
- [pironti2012identifying]
"Identifying website users by TLS traffic analysis -- New
attacks and effective countermeasures", n.d..
- [rahman18gan]
"Generating Adversarial Packets for Website Fingerprinting
Defense", n.d., <[https://www.rahmanmsaidur.com/projects/
Fall_18_Generating_Adversarial_Packets.pdf](https://www.rahmanmsaidur.com/projects/Fall_18_Generating_Adversarial_Packets.pdf)>.
- [rahman2018using]
"Using Packet Timing Information in Website
Fingerprinting", n.d..
- [rahman2019tik]
"Tik-Tok -- The Utility of Packet Timing in Website
Fingerprinting Attacks", n.d.,
<<https://arxiv.org/pdf/1902.06421.pdf>>.
- [reed2017identifying]
"Identifying https-protected netflix videos in real-time",
ACM on Conference on Data and Application Security and
Privacy, 2017 , n.d..
- [RFC7540] Belshe, M., Peon, R., and M. Thomson, Ed., "Hypertext
Transfer Protocol Version 2 (HTTP/2)", RFC 7540,
DOI 10.17487/RFC7540, May 2015,
<<https://www.rfc-editor.org/info/rfc7540>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol
Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018,
<<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS
(DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018,
<<https://www.rfc-editor.org/info/rfc8484>>.
- [rimmer2018automated]
"Automated website fingerprinting through deep learning",
Network & Distributed System Security Symposium (NDSS),
2018 , n.d..

- [schuster2017beauty]
"Beauty and the burst -- Remote identification of encrypted video streams", USENIX Security, 2017 , n.d..
- [shmatikov2006timing]
"Timing analysis in low-latency mix networks -- Attacks and defenses", European Symposium on Research in Computer Security, 2006 , n.d..
- [shulman2014pretty]
"Pretty bad privacy -- Pitfalls of DNS encryption", Workshop on Privacy in the Electronic Society, 2014 , n.d..
- [siby2018dns]
"DNS Privacy not so private -- the traffic analysis perspective", n.d..
- [sirinam2018deep]
"Deep fingerprinting -- Undermining website fingerprinting defenses with deep learning", arXiv preprint arXiv:1801.02265 , n.d..
- [sun2002statistical]
"Statistical identification of encrypted web browsing traffic", IEEE, 2002 , n.d..
- [tammam2012exploiting]
"Exploiting packet-sampling measurements for traffic characterization and classification", International Journal of Network Management, 2012 , n.d..
- [TIME]
"A Perfect CRIME? Only TIME Will Tell", Black Hat Europe 2013 , n.d..
- [trevisan2016towards]
"Towards web service classification using addresses and DNS", Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International. IEEE, 2016 , n.d..
- [wagner1996analysis]
"Analysis of the SSL 3.0 protocol", USENIX Workshop on Electronic Commerce Proceedings, 1996 , n.d..
- [wang2013improved]
"Improved website fingerprinting on tor", Workshop on privacy in the electronic society, 2013 , n.d..

- [wang2014comparing]
"Comparing website fingerprinting attacks and defenses",
Technical Report 2013-30, CACR, 2013. , n.d..
- [wang2014effective]
"Effective Attacks and Provable Defenses for Website
Fingerprinting", USENIX Security Symposium, 2014 , n.d..
- [wang2015walkie]
"Walkie-talkie -- An effective and efficient defense
against website fingerprinting", n.d..
- [wang2016website]
"Website fingerprinting -- Attacks and defenses",
University of Waterloo , n.d..
- [wright2009traffic]
"Traffic Morphing -- An Efficient Defense Against
Statistical Traffic Analysis", NDSS, 2009 , n.d..
- [yan2018feature]
"Feature selection for website fingerprinting",
Proceedings on Privacy Enhancing Technologies, 2018 ,
n.d..

Appendix A. Acknowledgements

The authors would like to thank Frederic Jacobs and Tim Taubert for feedback on earlier versions of this document.

Authors' Addresses

Ian Goldberg
University of Waterloo

Email: iang@uwaterloo.ca

Tao Wang
HK University of Science and Technology

Email: taow@cse.ust.hk

Christopher A. Wood
Apple, Inc.

Email: cawood@apple.com