# Poster: Improved Website Fingerprinting Attacks based on Tor Network Protocols


Loc Ho
Arkansas State University
Jonesboro, AR, USA
loc.ho@smail.astate.edu

Jongwook Lee
Dankook University
Yongin-si, South Korea
gomljo0307@naver.com

Won-gyum Kim
AiDeep
Seoul, South Korea
wgkim@aideep.ai

Donghoon Kim
Arkansas State University
Jonesboro, AR, USA
dhkim@astate.edu

Doosung Hwang
Dankook University
Yongin-si, South Korea
dshwang@dankook.ac.kr



***Abstract*—Tor network protocols enhance Tor network's privacy, security, and resilience to deanonymization attacks by exchanging information on many steps and data. The information, on the other hand, is commonly used in Tor network protocols. Network packets required for Tor protocols are exchanged at the beginning to support both onion services and general websites via Tor browsers. These packets are common in each category. To analyze how common Tor protocol packets affect the website fingerprinting attacks, we create the feature using network packets removed of the first several packets. Experimental results show that classification performance improves as the number of the beginning packets removed increases. This pilot study indicates that Tor network protocols designed to improve the security may be vulnerable to website fingerprinting attacks.**


## I. Introduction

The Tor (The Onion Router) is a Firefox-based anonymous network web service, with more than millions users worldwide via secure connection. Tor browser can access both general (non-hidden) websites and onion (hidden) services [1]. The non-hidden service uses a Tor browser to access a general website (e.g., CNN, google) while remaining anonymous using 3 relay nodes [2]. The hidden service is a website that can only be accessed by a Tor browser through Tor network, which uses the special-use top level domain (TLD) `.onion` instead of `.com`, `.net`, `.org`, etc.

Accessing hidden services with a Tor browser follows a different protocol than visiting general websites. Tor browsers do not receive messages directly from the hidden service; instead, they meet at a rendevous point chosen by a Tor browser [3]. This onion service protocol requires many steps and a lot of network packets to exchange information. These network packets can help to create features that enable to classify onion services and general websites. On the other hand, these packets are commin in each category (i.e., onion services and general websites).

The website fingerprinting attacks using machine learning have shown their validity and availability in terms of feature representation and generalization performance [4], [5]. The pilot study is to verity how the common network packets for Tor protocols affect the website fingerprinting attack depening on features. The contributions of this study are as following: (1) We describe the differences between the protocols of the non-hidden service and the hidden service over Tor network. (2) We investigate how Tor network protocol packets affect website fingerprinting attacks depending on how the features are structed.

## II. Related Work

Panchenko *et al.* [4] studied the practical limits of website fingerprinting at Internet scale with more than 300,000 webpages. They proposed the CUMUL feature using the number of cumulative sum of packet sizes and numbers. Yan and Jasleen [6] conducted an exhaustive feature analysis within eight different communication scenarios. They grouped features into five levels, such as packet, burst, TCP, Port, and IP address. They enumerated most informative features in each scenario. Lashkari *et al.* [7] used 23 features regarding time. They classified Tor or No-Tor data, and categorized 9 different applications.

## III. Research Approach

### A. Threat Model

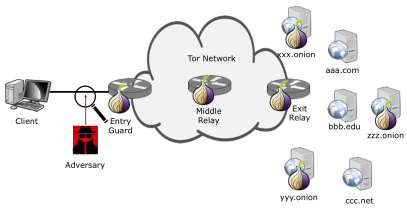


Fig. 1: The Threat Model

An adversary can monitor the network traffic from a client to the entry Tor router (entry guard) and from the exit Tor router to a destination client. Examples of adversaries may be a Tor router owner, ISP (Internet Service Provider), or local network administrator. In this work, we assume that an adversary monitors the network traffic between the client and the entry guard (i.e., the first router) as in Figure 1.

### B. Data Collection

We developed the network traffic collection system. The system works in the virtual environment (e.g., VirtualBox) to minimize network noise and has various features, such as traffic collection scenarios and GUI. The system has several options to collect the most suitable data for the real work environment. For example, the system can determine whether a site should continue to collect traffic and move to the next website, or collecting the sites on the lists once and then repeating them to multiple times. Because the contents of websites tend to change over time, the performance of ML based models may vary depending on the learned datasets [9]. Thus, it is important to collect the data that is best suited to the actual environment.

| Features | Category | Metrics | Number of removed packets | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | **0** | **10** | **20** | **30** | **40** | **50** | **100** |
| 125 features [8] | 30 general websites | Accuracy | 0.883 | 0.871 | 0.956 | 0.975 | 0.981 | 0.983 | 0.973 |
| | | Time | 5.055 | 9.888 | 21.601 | 32.835 | 45.317 | 58.215 | 73.625 |
| | 30 onion services | Accuracy | 0.593 | 0.580 | 0.801 | 0.863 | 0.887 | 0.896 | 0.883 |
| | | Time | 5.666 | 10.615 | 22.512 | 34.921 | 47.206 | 60.465 | 75.024 |
| 104 CUMUL [4] | 30 general websites | Accuracy | 0.914 | 0.914 | 0.904 | 0.898 | 0.886 | 0.877 | 0.858 |
| | | Time | 8.757 | 9.679 | 9.698 | 9.686 | 9.701 | 9.772 | 10.270 |
| | 30 onion services | Accuracy | 0.736 | 0.695 | 0.654 | 0.609 | 0.575 | 0.542 | 0.482 |
| | | Time | 10.783 | 10.865 | 11.178 | 11.450 | 11.644 | 11.629 | 11.533 |

TABLE I: Website fingerprinting attacks (Time: training time, Unit: second)

Data were collected from 30 onion services that are compiled in `ahmia.fi` [10]. 28 onion services are V2 and 2 onion services are V3 which reinforces the Tor network's security[1]. In addition, 30 general websites were collected for comparison with onion services. Each service has 150 instances.

### C. Feature vectors

This work uses two previous works as feature vectors. The first work is CUMUL with 104 features [4]. The 104 CUMUL consists of 4 basic features (the number of incoming/outgoing packets and the sum of incoming/outgoing packets) and 100 cumulative sum of packets through linear interpolation. The second work is 125 features [8]. The 125 features consist of various network information, such as packet inter arrival time, packet ordering, burst time, etc.

## IV. Analysis

To investigate how Tor network protocol packets affect website fingerprinting attacks in view of how the features are created, the first several packets are removed before we create two features (125 features [8], and 104 CUMUL [4]) The classification was conducted with Decision Tree (DT), Random Forest (RF), Extra Tree (ET), and XGBoost (XGB). The results are presented in Table I with only RF due to the limited space in this publication. The overall result is that general websites are better than onion services in both features based on this experimental environments. When the packets are not removed, 104 CUMUL shows a slightly better result than 125 features; for 30 general websites, the 104 CUMUL's accuracy is 91.4%, while the 125 feature's accuracy is 88.3%. When the packets are removed, however, it can be observed that the accuracy for 125 feature improves while the accuracy for 104 cumul decreases. In 125 features of 30 general websites, the accuracy improves as more packets are removed, reaching 98.3% after 50 packets have been removed. On the other hand, even when packets are removed in CUMUL, the accuracy tends to decline slightly; the accuracy is 91.4% without removing and the accuray 87.7% when 50 packets are removed.

According to our experimental results, we can find the following. Tor network protocols require common packets for improving the Tor network's privacy, security, and resilience to deanonymization attacks. The website fingerprinting attacks, however, are effective when the common packets can be removed. Packets used to enhance the security of the Tor protocol may not help prevent fingerprinting attacks on Tor network.

[1] https://gitweb.torproject.org/torspec.git/tree/rend-spec-v3.txt

## V. Conclusion

This pilot study showed how website fingerprinting attacks can be improved based on Tor network protocols. From the experimental analysis, the Tor network protocol can be vulnerable to website fingerprinting attacks due to common network packets. We are experimenting with how many packets are commonly used to support onion services and general websites for our future work.


## Acknowledgment

This material is based upon work supported by the National Science Foundation under Award No. OIA-1946391.


## References


[1] Philipp Winter, Anne Edmundson, Laura M Roberts, Agnieszka Dutkowska-Żuk, Marshini Chetty, and Nick Feamster. How do tor users interact with onion services? In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 411–428, 2018.

[2] Tao Wang and Ian Goldberg. Improved website fingerprinting on tor. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, pages 201–212, 2013.

[3] How do onion services work? https://community.torproject.org/onion-services/overview/. Accessed on January 8, 2021.

[4] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *NDSS*, 2016.

[5] Rebekah Overdorf, Mark Juarez, Gunes Acar, Rachel Greenstadt, and Claudia Diaz. How unique is your. onion? an analysis of the fingerprintability of tor onion services. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2021–2036, 2017.

[6] Junhua Yan and Jasleen Kaur. Feature selection for website fingerprinting. *Proc. Priv. Enhancing Technol.*, 2018(4):200–219, 2018.

[7] Arash Habibi Lashkari, Gerard Draper-Gil, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of tor traffic using time based features. In *ICISSp*, pages 253–262, 2017.

[8] Donghoon Kim, Loc Ho, Young-Ho Kim, Won-gyum Kim, and Doosung Hwang. Poster: A pilot study on real-time fingerprinting for tor onion services. In *The Network and Distributed System Security Symposium (NDSS) 2021*, 2021.

[9] Hyungseok Oh, Donghoon Kim, Won-gyum Kim, and Doosung Hwang. Performance analysis of tor website fingerprinting over time using tree ensemble models. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI 2020)*, 2020.

[10] Tor hidden service search. https://ahmia.fi. Accessed on January 8, 2021.