UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft
Fakultät für Elektrotechnik, Informatik und Mathematik
Arbeitsgruppe IT-Sicherheit

Seminar Thesis

Website Fingerprinting Defense: Walkie Talkie — A Review

Ashwin Prasad Shivarpatna Venkatesh

Date: February 22, 2019
Advisor: Peter Chvojka

Abstract

Website fingerprinting (WF) attack is an attempt made by a passive, local adversary to identify the user's website accesses by leveraging implicit properties of network traffic flow. Academic work has shown that WF attacks are effective even when the user is explicitly using privacy preserving technologies like Tor. A number of WF defenses have been proposed and also counter attacks to defeat these defenses are published.

In this paper, we discuss a lightweight and efficient WF defense, Walkie-Talkie (W-T). W-T is designed to defend against any website fingerprinting attack by carefully masking the traffic flow and hiding the unique features of a website available to the eavesdropper. W-T utilizes Half-duplex communication to make the traffic molding efficient. W-T is by far the best WF defense as it has a good balance between overhead and effectiveness. We also discuss the state-of-the-art attacks on W-T, highlighting that W-T is still unbeaten.

Contents

1 Introduction

Privacy has always been a concern for users on the internet and users tend to use privacy enhancing tools such as virtual private networks and Tor to hide their activity from unauthorized parties. Modern routing techniques are built on the principle of preserving privacy. Tor[1] is one such project which is used extensively in recent times, Tor promises anonymity by utilizing nested levels of encryption. However, a local adversary who can monitor the traffic between the client and server, is shown to be capable of determining the client's activity with a fair amount of accuracy and this is possible because the adversary has access to the implicit signature of the traffic such as packet length, quantity, direction, timing and many more which are collectively referred to as Website Fingerprint (WF). By analyzing such properties and extracting patterns, it is possible to estimate the web page the user accessed, irrespective of the precautions taken by the user.

In this paper, we discuss "Walkie Talkie" (W-T), a defense mechanism that transforms the traffic signature such that, the adversary can only see information that can be classified in at least two different ways, meaning the accuracy of the adversary to identify the page visited by the user is reduced to random guessing. Apart from this, we also discuss state-of-the-art attacks against W-T.

In Chapter 2, we introduce the terms and technologies used in the WF domain as priori and a brief discussion on the academic efforts in the WF domain. In Chapter 3, we discuss the approach taken by the authors of W-T to defend against the WF attacks. The implementation details are in the Chapter 4. W-T evaluation against other WF attacks and defenses are in Chapter 5. We discuss state-of-the-art academic work related to W-T and WF in the Chapter 6.

[1] `https://www.torproject.org/`

2 Background

In the Section 2.1, we explain the basics of onion routing, which is a key to follow the rest of the paper. Section 2.2 establishes the threat model and assumptions about the attacker, followed by a discussion on the previous academic work in the WF domain.

2.1 Onion Routing

Onion Routing is an overlay network mechanism to enable anonymous internet access. Tor is a popular implementation of Onion routing which is run and maintained by volunteers across the world. In Tor, clients establish a circuit in the network which consists of a minimum of 3 nodes. The user traffic flows through the circuit in fixed-size cells, each of which is 512 bytes that includes a header and payload. Cells are either (1) Relay cells or (2) Control cells based on the command in the header. Control cells are read by the node that receives it and the relay cells are used to carry end-to-end data. Payload in the replay cells are encrypted multiple times using 128-bit AES to form layered encrypted message. Client constructs the circuit incrementally by negotiating a symmetric key with each of the selected nodes using Diffie–Hellman key exchange.

Once the circuit has been established, the client can send relay cells with encrypted data. Let's say Alice wants to send a message to Bob through a circuit with 3 nodes. An overview of this end-to-end message exchange can be described as follows, Alice who has already exchanged symmetric keys with all the nodes, applies multiple layers of encryption on the message incrementally with respect to the node order in the circuit by using 128-bit AES in counter mode. The encrypted relay cell is passed on to the entry node of the circuit, where each node decrypt one layer of encryption and forwards it to the next node in the circuit or operate on the instructions if it is the final node in the circuit, thereby forwarding the message to Bob.

While the precautions are taken by the client to preserve his privacy, Tor is still vulnerable to man-in-the-middle attack where the local adversary can monitor the cell signature and use website fingerprinting techniques to realize the online activity of the client.

2.2 Attacker model

Capabilities and assumptions made about the attacker are listed below and most of these assumptions are common to all the previous work in the WF domain.

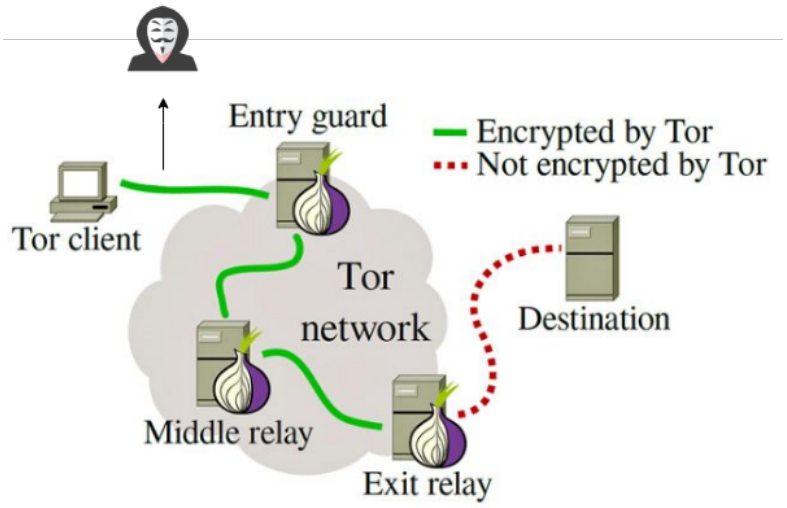

Figure 2.1: Local, Passive attacker

Local and Passive attacker Passive attack is where the attacker is only an active observer and does not actively involve in altering the data. A passive attacker in the context of this paper is someone who is capable of monitoring the traffic between the client and the Tor entry node, this could be a local attacker connected on the same local network, network administrator or the internet service provider.

Closed and Open world set An attacker will monitor and learn patterns in the cell flow signatures for a particular set of websites from his own visits, such

a selection of known websites are referred to as the monitored/closed world set. The goal of the attacker is to capture the cell flow statistics of the client and then deduce the website a client is visiting within the closed world set, by comparing the signatures captured previously. Academic literature show that, by using machine learning techniques, the accuracy with which the client's activity can be predicted is as high as 98% in some cases [SIJW]. WF attacks on open world is not feasible yet and the evaluation of such scenarios is discussed in the later sections.

Homepage fingerprinting Many academic literature in the WF domain consider only the homepage signature in their work which means the internal page accesses are not fingerprinted and used for analysis, which is debated to be not practical, but in the context of Walkie Talkie, only the homepage signature is considered. This is only to compare walkie talkie defense against older attacks, otherwise W-T is designed to defend any open world page.

Single tab browsing The client is assumed to be browsing only a single page at a time, so that the traffic observed between the client and the entry node belongs to a unique page. Most attacks on WF make this assumption. It is either this assumption or as an alternative, the attacker should be able to filter traffic originating from a certain page from all other traffic.

2.3 Academic work in WF

Walkie Talkie defense itself was presented in Usenix 2017. There are many WF attacks and defenses published, both before and after the publication of W-T. In this section, we will discuss some of the attacks and defenses

2.3.1 Attacks

Attacks on privacy using WF has the same fundamental principle that attackers try to learn and classify websites based on traffic signatures (Fig 3.1). Tor inherently has a mechanism to protect against WF attacks, as Tor transmits messages in fixed cells (Section 2.1). As a result, the packet length information cannot be used as a signature to classify pages in Tor. WF Attacks that relied on such metrics failed

to achieve significant accuracy of website identification [WG]. However, attacks that utilized more relevant metrics to train the machine learning classifiers have been achieving better results and the state-of-the-art techniques have reached an accuracy of more than 90% [SIJW]. To train the classifiers these techniques rely on metrics such as cell ordering, cell direction, edit-distances, incoming-outgoing cell quantity and cell burst sequences. We will be briefly discuss state-of-the-art WF attacks in Chapter 6.

An important consideration is the distinction between closed-world and open-world attack scenarios. In a closed-world situation, the packet sequence to be classified by the attacker is assumed to be from a known list of websites the attacker is actively monitoring. In the open-world scenario, the packet sequence to be classified can originate from outside the known list. Most previous work in WF attacks are successful only in the close-world setting, however, newer attacks might be able to exploit implicit loop holes in WF, thus posing a practical privacy problem.

2.3.2 Defenses

Similar to earlier attacks, some WF defense focused on the packet length feature where the strategy was to transform the traffic such that it has uniform packet length. As the attacks incorporated more features, older defenses became less effective. However, the fundamental idea still remains the same, which is to morph the traffic such that the attacker's classification technique cannot effectively classify the observed signature as unique to one web page. Recent techniques can be weighed and compared based on the latency and bandwidth overhead it causes. The bandwidth overhead is the number of dummy cells added compared to the undefended case. The time overhead is the difference in time taken to load a web page compared to undefended scenario. We discuss the evaluation in detail in Chapter 5.

Tamaraw and BuFLO are some example defenses that morph traffic features and both these techniques has a overhead of at least 130% on bandwidth and run 2-4x slower compared to unprotected Tor browsing. Two modern techniques achieve WF defense with acceptable overhead of about 30-60%. **1) Walkie Talkie:** utilizing half-duplex communication (Section 3.1) and burst molding (Section 3.2). W-T achieves good results with relatively acceptable overhead. We will discuss the W-T implementation and evaluation in detail in Chapters 4 and 5. **2) WTF-PAD** uses a technique called adaptive padding, which masks the traffic signature. However, the bandwidth overhead is 20-30% more than W-T. These two techniques

are currently the best candidates to be adopted by the Tor project.

The goal of W-T is to protect against all previous and future classification based WF attacks by carefully molding the traffic such that a traffic signature looks like it might be originating from at least two different sources. W-T is still undefeated and the state-of-the-art attacks that rely on advanced machine learning techniques, such as Deep Fingerprinting [SIJW] can still only reach close to the theoretical maximum of 50% proved by W-T authors Wang and Goldberg [WG]

3 Approach

The mechanism for WF attack depends on the classification techniques, while the goal of WF defense is to hide the unique characteristics exposed by the traffic signature. In this chapter we will examine the approach taken by W-T to equalize the traffic flow. The key idea in Walkie Talkie is to use Half-duplex communication along with burst molding to transform a cell sequence into burst sequence which cause collisions in the attackers classification technique. These two concepts are explained in the following sections.

3.1 Half-duplex communication

In a typical internet browser working in full-duplex mode, the requests are made continuously without a rule being enforced on the ordering of requests made to the server. The browser is designed to make requests as soon as possible without waiting for the previous requests to be fulfilled. In W-T, Wang and Goldberg [WG] utilized Half-duplex communication (HDC), where the client browser only sends further requests when all the previously made requests are completed and thereby leading to a grouped interleaving of incoming and outgoing cell bursts (Figure 3.1). A brief overview of the implementation in Tor browser is discussed in the Chapter 4. The key concepts behind using burst sequences are listed below, which are also some of the reasons why W-T has achieved considerably lower overheads compared to other defenses.

- Reduction of information available to the attacker from cell sequence to burst sequence.
- Optimization of sequence molding. Molding of burst sequences is less overhead when compared to cell sequence molding.
- Reduction in the meta-data needed by the client for sequence molding.

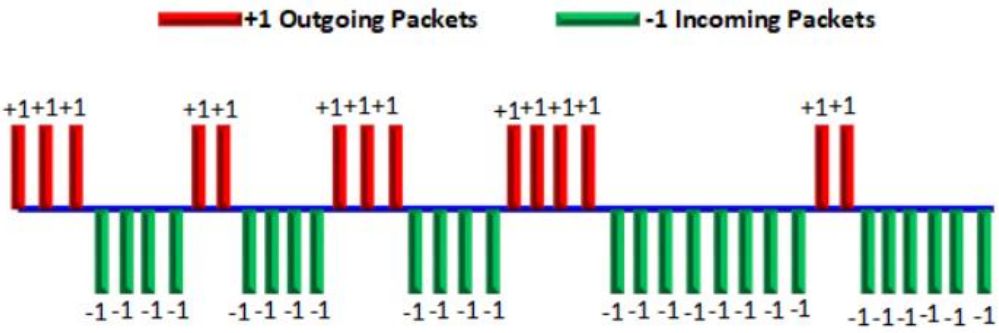

Figure 3.1: Example burst sequence [Rah]

3.2 Burst molding

Earlier academic work in WF defense masked the unique cell signature of a sensitive site by simultaneously loading a decoy page, thereby confusing the attacker's classification algorithm which cannot separate the cells to be originating from one of the two sites and extract a unique signature. This strategy is straight forward and will be effective irrespective of the classification technique used by the attacker. However, the problem is that, loading a decoy page causes approximately 100% bandwidth overhead. [WG]

Two works, Super-sequences and Glove used a mechanism to only simulate loading of two pages to limit the overhead. This is achieved by loading the super-sequence of the two pages. s' is a super-sequence of s if s' contains s. A super-sequence is constructed by adding fake cells to the original sequence, this is called burst molding. The procedure followed by W-T for burst molding is as follows. (Figure 3.2)

- Number of cells in Real page: $b_i = (b_{i+}, b_{i-})$
- Number of cells in Decoy page: $b'_i = (b'_{i+}, b'_{i-})$
- Molded sequence: $\hat{b}_i = (\ max\{b_{i+}, b'_{i+}\},\ max\{b_{i-}, b'_{i-}\}\ )$

b_{i+} : Outgoing burst sequence, b_{i-} : Incoming burst sequence.

If the number of bursts in the real and decoy sequences are not equal, then, entirely

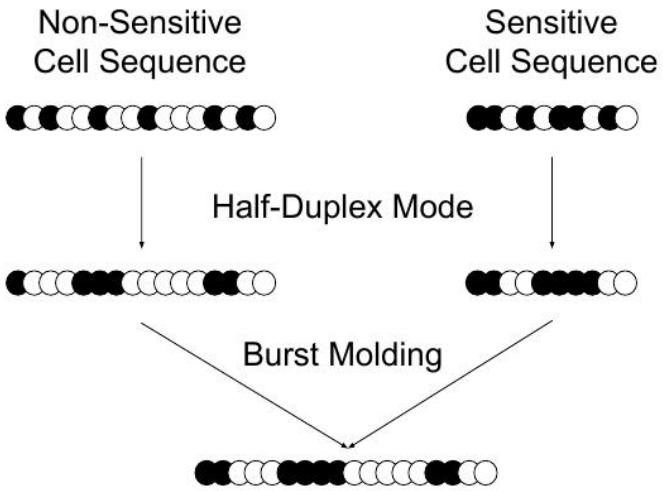

Figure 3.2: Burst molding example [WG]

fake bursts are added to balance the shorter sequence. It is to be noted that, fake cells add to bandwidth overhead but not time overhead as the fake cells are dropped by the proxy. While the fake bursts add to both the bandwidth and time overhead as the extra time is utilized to process and drop the whole burst without sending any real cells.

3.3 Advantages

Burst molding and HDC have certain advantages and helps W-T in achieving a lower overhead compared to other defenses, they are listed below.

- Ability to choose the decoy page used to build the super-sequence.
- W-T simply uses the max of two burst sequences as seen in the previous section, this avoids complicated computation.
- Usage of burst sequences instead of cell sequences for molding, which is 1800 times more efficient to store and use. [WG]
- Possibility of further optimization of burst molding by pre-computing the best decoy page for a set of sensitive pages.

4 Implementation in Tor browser

In this chapter, we explore how the implementation of half-duplex and burst molding could work in practice. In W-T, Wang and Goldberg [WG] demonstrated a proof-of-concept by implementing half-duplex and burst molding in the Tor client.

Wang and Goldberg [WG] implemented HDC by adding two states to the Tor browser, ***1) Walkie:*** Idle browser and ***2) Talkie:*** Actively loading a page. In the talkie state, the browser will not load any further requests, the requests will instead be queued.

The browser needs to establish a TCP connection before sending a HTTP request and as this happens in two phases, the delay between them can cause the browser to send a second request when the server is already communicating and thus violating HDC. To solve this authors utilize an existing technique called *optimistic data*, where both the connection and HTTP requests are sent in immediate succession as the connection is pretended to the browser to be established.

The super-sequence is constructed together by both the client and the proxy. Proxy in the Tor case can be the entry node itself. Authors added two new cell types to Tor, a "fake cell" and a "fake burst end cell" to facilitate super-sequence construction. Super-sequence is generated by adding fake cells and bursts to the original cell sequences. Client first chooses a decoy burst sequence and communicates this to the proxy before starting the real page visit. The proxy itself is only counting the number of packets sent in each of the bursts and adds fake cells if it is lower than expected. Hence, the computational load on the proxy is acceptable.

During a real page visit, the client sends the fake cells first and then the remaining real cells. The proxy will drop the fake cells and similarly return the responses by starting to send fake cells before real cells. But, when the need to send entirely fake bursts arise, fake burst end cells are used to mark the end of each fake burst. A queue is maintained when a fake burst is in process, each cell created in between fake bursts will be queued and sent as the next burst.

5 Evaluation

The effectiveness of W-T can be realized by comparing its performance against state-of-the art WF attacks and defenses.

The data collection is made using the modified Tor browser as discussed in the previous chapter. Burst molding is only simulated as the authors investigated a large number of parameter choices. Evaluation is based on the closed-world data set of the top 100 pages on Alexa[1], this is to make a reasonable comparison to all the previous work in the WF domain which use the same methodology.

In Section 5.1, W-T is compared to other attacks in terms of attacker accuracy and in Section 5.2, W-T is compared with other WF defenses to highlight the low overhead achieved.

5.1 W-T vs Attacks

Closed-world accuracy

In the Table 5.1, state-of-the art WF attacks are listed along with the accuracy of classification in the *undefended* case, where the packet sequence is unaltered and *defended* case, where the traffic is molded using W-T. Data in the Table 5.1 is aggregated from the works of Wang and Goldberg [WG] and Sirinam et al. [SIJW].

W-T is effectively lowering the accuracy of all attacks to under 0.5, which makes the attacks no better than random guessing. It can be seen that, even after two years of initial publication, W-T still is effective as of today against advanced attacks such as Deep Fingerprinting (DF).

[1] `https://www.alexa.com/`

Attack	Undefended	Defended
kNN	0.95	0.28
SVM	0.81	0.44
CUMUL	0.64	0.20
DF	0.98	**0.49**

Table 5.1: Closed-world accuracy of WF attacks against W-T

Open-world accuracy

Wang and Goldberg [WG] evaluated the open-world scenario by studying the true positive rate (TPR) and false positive rate (FPR) of various attacks as shown in the Table 5.2. The effectiveness of W-T can be seen from the Table 5.2, TPR is significantly reduced and FPR is increased for each attack. According to the base rate fallacy, these results could lead to incorrect interpretation of data and the actual accuracy can be much lower [WG][SIJW]. Due to the base rate fallacy, Sirinam et al. [SIJW], decided to evaluate the open-world accuracy in terms of precision-recall curves and found that all attacks perform poorly against W-T, with DF having a comparatively better precision of 0.36. For further information, we would refer readers to the main work "Deep Fingerprinting" by Sirinam et al. [SIJW].

	True Positive Rate		False Positive Rate	
Attack	**Undefended**	**Defended**	**Defended**	**Defended**
kNN	0.98	0.68	0.09	0.62
SVM	0.47	0.33	0.05	0.20
CUMUL	0.78	0.20	0.04	0.35

Table 5.2: Open-world accuracy of WF attacks against W-T

5.2 W-T vs Defenses

Overhead in the context of W-F is measured in two aspects, **1) Bandwidth overhead (BWOH):** is the ratio of dummy cells added by the defense and the number of cells in the original cell sequence. **2) Time overhead (TOH):** is the

ratio of excess time taken to load the cell sequence to the total time required to load the original sequence.

It can be seen from the Table 5.3 that the overhead incurred by W-T is significantly lower compared to other defenses which provide similar degradation in attacker accuracy. Data from Table 5.3 is also aggregated from the works of Wang and Goldberg [WG] and Sirinam et al. [SIJW] to showcase the statistics from state-of-the-art.

Another important observation to be made in the Table 5.3 is that DF attack comes close to the theoretical maximum attacker accuracy proved by W-T.

	Overhead		**Accuracy of WF attacks**	
Defenses	**BWOH**	**TOH**	**DF**	**kNN**
BuFLO	246%	137%	12.6%	10.4%
Tamaraw	328%	242%	11.8%	9.7%
WTF-PAD	64%	**0%**	**90.7%**	16.0%
Walkie-Talkie	**31%**	**34%**	**49.7%**	20.2%

Table 5.3: W-T compared to other WF defenses

6 Related Work

Some of the recent academic work that is closely related to and mentions W-T in their work are discussed here. All of them are published in the year 2018.

Sirinam et al. [SIJW] implemented Deep Fingerprinting, which is a state-of-the-art WF attack devised against Tor networks. This work is extensively used in this paper to compare against the latest attempts to break W-T defense. DF leverages advanced machine learning mechanisms such as deep learning to train the data-set of packet sequences. DF achieves a high accuracy of 98% on undefended Tor traffic, which is the highest compared all previous attacks.

Sirinam et al. mainly highlight the fact that Tor traffic is not as safe as promised and the need to employ an efficient WF defense. The outcome of this paper also helps in picking the right candidate for implementing WF defense in Tor. WTF-PAD and Walkie Talkie were the two main contenders due to low overhead and effectiveness, but DF defeats WTF-PAD with 90% accuracy in closed world scenario. However, Authors conclude the paper by noting the challenges to implement W-T in practice and this is summarized in the following list.

- The need of a directory server to take responsibility of collection, maintenance and distribution of burst sequences of websites for burst molding.
- Additional context (such as user language, etc.) should be available to the client browser to perform efficient selection of decoy pages for burst molding. Which could be hard to implement into a system.
- Unlike WTF-PAD, W-T induces 31% TOH. This is on top of already slow Tor networking.

Bhat et al. [**BLKD**] introduced DynaFlow, a WF defense which also does burst molding, but unlike W-T, DynaFlow tries to do constant molding for all

traffic. Bhat et al. argue that the construction of super-sequences is not practical as it requires the client to have a database of packet sequences which are to be constantly updated, however the overheads they incur is significantly higher than W-T. Bhat et al. also propose a WF attack called "Var-CNN", with an automated way of extracting features from packet sequences and utilizes neural networks.

Cui et al. [**CYGCT**] is another WF defense which has achieved much lower overhead compared to W-T, at 0% TOH and 20% BWOH. Cui et al. achieve this by implementing a noise generation algorithm, which instead of generating random noise, generates realistic noise based on the user traffic history. The idea is based on the inability of the attacker to classify packet sequences when the client is visiting multiple websites at the same time.

Rahman [**Rah**] in his master thesis, experimented on utilizing packet timing information as features for a WF attack based on convolution neural networks. Author achieved 47% accuracy and expects to further increase by training of a bigger data-set.

Rimmer et al. [**RPJ**$^+$] developed another WF attack based on deep learning, which utilized the biggest WF dataset to train their classifier, which also depended on automated extraction of features. Evaluation against W-T is not made but only mentioned as a future work.

Conti et al. [**CLMS**] recently made a survey on the state-of-the art network surveillance techniques and mentioned W-T as a considerable countermeasure for mitigating traffic analysis as W-T provides an acceptable trade-off between BWOH and TOH.

7 Future Work & Conclusion

We discussed Walkie-Talkie, an efficient, low overhead defense against all website fingerprinting attacks. W-T implements half-duplex communication in the browser and burst molding to manipulate the traffic signature with minimal overhead. W-T is also safe against all possible WF attacks which work on the principle of finding patterns in traffic sequences, as W-T molds the traffic sequence to appear as-if it could be originating from at least two different sources. We also notice that state-of-the-art website fingerprinting attacks still fail to defeat W-T, although the recent attempts have achieved 49% accuracy which is close to the theoretical maximum proven by W-T.

W-T is currently the best candidate to be incorporated by the Tor project even with some practical limitations we have discussed. A close competitor is WTF-PAD with a better time overhead, but recently proven to be ineffective against advanced WF attacks.

WF is still an open problem and further research is required to mitigate this efficiently. To further improve the defenses, advanced machine learning techniques can be used to dynamically create dummy traffic sequences according to the user context to confuse the attacker. The main challenge is still the overhead caused by any of the approaches taken by the WF defenses and a bigger challenge is the large scale adoption by distributed technologies like Tor, so that the end user is benefited.

Bibliography

[BLKD] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. Var-CNN and DynaFlow: Improved attacks and defenses for website fingerprinting.

[CLMS] M. Conti, Q. Q. Li, A. Maragno, and R. Spolaor. The dark side(-channel) of mobile devices: A survey on network traffic analysis. 20(4):2658–2713.

[CYGCT] W. Cui, J. Yu, Y. Gong, and E. Chan-Tin. Realistic cover traffic to mitigate website fingerprinting attacks. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1579–1584.

[Rah] Mohammad Saidur Rahman. Using packet timing information in website fingerprinting.

[RPJ+] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated website fingerprinting through deep learning. In *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society.

[SIJW] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning.

[WG] Tao Wang and Ian Goldberg. Walkie-talkie an efficient defense against passive website fingerprinting attacks. page 16.