Website Fingerprinting on LEO Satellite Internet

by

Prabhjot Singh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2023

© Prabhjot Singh 2023

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Although encrypted channels, like those provided by anonymity networks such as Tor, have been put into effect, network adversaries have proven their capability to undermine users' browsing privacy through website fingerprinting attacks.

This study examines the susceptibility of Tor users to website fingerprinting when data is transmitted via Low Earth Orbit (LEO) satellite Internet connections. To this end, we design an experimental testbed that includes a Starlink satellite Internet connection and a traditional fiber connection. We use this testbed to gather Tor browsing data over both LEO and fiber connections, enabling a study over the effectiveness of website fingerprinting attacks in these different settings. Besides using our testbed to gather Tor traces, we also collect simple website accesses via Firefox in order to characterize Tor and non-Tor traffic in both Starlink and fiber network settings.

We were able to observe clear differences between Starlink and fiber connections for both Tor and non-Tor traffic when analyzing metrics such as average page load time, average number of packets, and average length of packets. Ultimately, our research leveraging state-of-the-art website fingerprinting attacks suggests that Tor traffic transmitted through Starlink is just as susceptible to these attacks as traffic over fiber links, despite the unique networking characteristics of Starlink connections. However, we find out that the deployment of website fingerprinting defences can substantially decrease the effectiveness of these attacks on Tor traffic exchanged via Starlink, resulting only in a slight bandwidth usage overhead when compared to the deployment of the same defenses in fiber connections.

Acknowledgements

First and foremost, I extend my deepest gratitude to Waheguru Ji (God), for providing me with the strength, wisdom, and serenity needed throughout my research journey and the writing of this thesis.

I would like to express my heartfelt thanks to my family. To my dear parents, whose love and guidance have been my constant light in the darkest of times, and to my brother, whose support and encouragement have been unwavering. Your faith in my abilities has been the bedrock of my resilience and determination.

Special thanks go to my supervisor, Diogo Barradas, for his invaluable guidance, patience, and expertise. Your mentorship has been instrumental in shaping both my research and my professional growth. I am deeply appreciative of the knowledge and insights you have shared with me.

I am also immensely grateful to Lori Paniak for his assistance and support in the experimental setup.

In closing, I thank everyone who has been a part of this journey, directly or indirectly. Your support has been a source of motivation and inspiration, and for this, I am eternally grateful.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The Internet has experienced a significant growth since its inception, currently encompassing a user base of approximately 5.3 billion individuals [44]. To ensure that Internet users can communicate securely in face of network adversaries with the capabilities to intercept their communications, encryption protocols like TLS (and its predecessor, SSL) [49] prevent adversaries from eavesdropping or manipulating the contents of users' data exchanges. The use of encryption protocols has then enabled the growth of a multitude of security-sensitive applications, including e-banking, e-commerce or tele-health [39].

Despite the benefits provided by encryption to the Internet ecosystem, the simple activity of web browsing can pose a number of threats to users' privacy. In fact, while encryption obscures the content of communications, network adversaries may still discern privacy-sensitive information about users, simply tracking the sequence of web pages a user visits over time. This means that, for instance, adversaries can exploit this information to gain insights into a user's health status or financial situation [59]. The main reason why these attacks are possible is because widespread encryption protocols like TLS do nothing about hiding communication metadata, such as the source and destination IPs of a given data exchange or the times at which these exchanges take place.

To shield themselves from the above risks, savvy Internet users are typically found to resort to privacy-enhancing technologies, like the Tor anonymity network [14], to conceal the identity of the websites they access through the Internet. Specifically, Tor makes use of a technique known as onion routing to obscure the destination IP address of a user's

communication by routing the user's traffic through multiple Internet nodes (or relays, usually three) that comprise a Tor circuit. Thus, adversaries that eavesdrop a user's Internet connection can easily perceive that the user is connected to a relay on the Tor network, but not what their final destination is.

Even though Tor provides an enhanced level of privacy to its users, recent studies have shown that network eavesdroppers can still overcome Tor's protections. Put briefly, an attacker can build a database of website fingerprints, i.e., a set of signatures drawn from different characteristics of the traffic observed when accessing a given website over Tor, and then attempt to match the traffic patterns generated by a Tor user to the fingerprints comprising this database [47]. Typically, the traffic characteristics used to build each website fingerprint consist of timing and direction characteristics of traffic, while the matching step usually depends on the application of a machine learning-based classifier that is trained using the fingerprints contained in the adversary's database.

In spite of the risks posed by website fingerprinting attacks, their accuracy is known to be sensitive to the underlying conditions of the network segments under analysis [25, 12], such as the available bandwidth, jitter, or packet drop rates. Thus, in the past, researchers have wondered whether (and to what extent) the risks of website fingerprinting attacks would transfer to other networking mediums with substantially different transmission characteristics, like wireless LTE/4G networks [28, 52]. These studies have shown that attackers could still be able to accurately fingerprint users' traffic over Tor in such settings.

More recently, we have assisted to an increasing prevalence of satellite networking solutions, powered by the launch of LEO (Low Earth Orbit) satellite network constellations like Starlink [57] or OneWeb [38]. These solutions have largely facilitated the provisioning of Internet access to users residing in remote regions of the world, and continue to be enhanced through the launch of more capable equipment [58] and provider-side upgrades to routing algorithms within the constellations themselves [4]. It remains unclear, however, what implications these recent satellite networking environments may have to the privacy of users. In the one hand, they promise connectivity speeds similar to fiber networks (or even faster), while on the other hand they make use of different wireless mediums which may be prone to several sources of interference [33, 66, 27, 48]. For instance, the latency and throughput of satellite connections are highly dynamic and users frequently experience service interruptions as a result of high packet loss [33]. Many environmental factors, including temperature, precipitation, cloud cover, solar storms, and terrain, have a significant impact on the performance and power consumption of satellite connections [33, 27]. These alterations in network performance have an effect on network traffic patterns and, consequently, may impact the website fingerprinting capability.

In this thesis, we aim to answer the question of whether LEO satellite Internet users are more vulnerable to website fingerprinting attacks than users using traditional fiber connections. To answer this question, we set up an experimental testbed using both a fiber and Starlink connection, and use it to collect a dataset of website accesses over Tor. We leverage state-of-the-art website fingerprinting attacks over our collected traces to understand whether network adversaries able to inspect the ground links of both kinds of connections are able to identify which websites are being accessed by users. Lastly, we evaluate the security benefits and performance trade-offs of existing defences when applied to fiber and satellite Internet links.

Our findings suggest that Tor traffic exchanged over Starlink Internet links is equally vulnerable to website fingerprinting attacks as Tor traffic exchanged over traditional fiber links. We hypothesize that, despite the different connectivity characteristics of the ground-satellite or traditional fiber links that connect our measurement node to the Tor network, most of the observed changes are absorbed and minimized by the network effects (e.g., added latency, jitter) caused by Tor's own circuitry.

1.2 Contributions

Towards carrying out the investigation laid-out in the previous section, this thesis provides the following list of technical contributions:

- The implementation of a laboratory testbed that includes a Starlink satellite dish. We leverage this testbed to collect a new dataset of Tor traffic over LEO satellite links. We also collect a new dataset over a fiber link to enable the direct comparison of results throughout.
- An experimental study over the success of state-of-the-art website fingerprinting attacks over satellite links.
- An exploration over the suitability of existing website fingerprinting defences to be deployed on LEO satellite-based Internet links.

1.3 Thesis Outline

This thesis is organized as follows. In Chapter 2, we provide background knowledge on Tor and website fingerprinting attacks/defences, as well as on recent satellite networking envi-

ronments. In Chapter 3, we describe the methodology of our study, including a description of our experimental testbed and data collection procedures. Chapter 4 presents our study on the susceptibility of LEO satellite Internet links to website fingerprinting attacks, where we also benchmark existing defences when deployed over satellite Internet links. Lastly, in Chapter 5, we summarize our main takeways, detail the limitations of our study, and point towards compelling directions for future work.

Chapter 2

Related Work

In this chapter, we start by providing background on the Tor anonymity network (Section 2.1), and explain how it provides Internet users with defences against surveillance. Then, we describe the typical setup used by adversaries to launch website fingerprinting attacks (Section 2.2), also surveying a body of existing attacks and defences. Finally, we detail past website fingerprinting attempts targeted at mobile networks' equipments and describe the satellite networking environment, which has so far been unexplored in the context of website fingerprinting (Section 2.3).

2.1 Background on Tor

The Tor network is an anonymous communication network that operates on a circuit-based system, utilizing a modified version of onion routing [14]. The primary objective of Tor's implementation of the onion routing protocol is to ensure the anonymity of senders utilizing TCP-based applications, specifically those engaged in web browsing activities. The Tor network relies on nodes, specifically relays or Onion Routers (OR), that are operated by volunteers. These nodes facilitate the forwarding of traffic along a circuit. Circuits are typically comprised of three relays, namely an entry relay, a middle relay, and an exit relay (also referred to as a node). When constructing circuits, clients engage in the process of relay selection by choosing relays from a list of options that are available to them. The aforementioned list can be accessed through specialized relays that function as directory authorities. Within the context of a circuit, it is pertinent to note that onion router relays possess knowledge solely pertaining to their immediate predecessor and successor relays, with no awareness of any other relays within the circuit. The transmission of data occurs

within circuits, wherein the data is organized into cells of a fixed size (512 bytes). These cells are encrypted using symmetric keys that have been previously shared with clients. Every individual cell is designated for a specific relay. There are two distinct types of cells, namely relay cells and control cells. Relay cells include end-to-end data, whereas control cells are interpreted by the OR that receives them (e.g., extend circuits). Relays possess the capability to multiplex numerous TCP streams concurrently within each circuit, thereby enhancing both efficiency and anonymity. Relay cells are equipped with end-to-end integrity checking checksums, enabling the final relays to discard any defective cells.

The Tor network employs a limited set of relays that function as directory authorities. They maintain a comprehensive record of the current Tor network status, which includes compiling relays, their corresponding certificates, and public keys. The document can be obtained by clients from any directory authority in order to acquire a comprehensive list of relays that are currently accessible. Subsequently, Tor implemented the inclusion of directory caches, which acquire a duplicate from directory authorities. On a regular basis, onion routers engage in the process of signing and subsequently publishing their router descriptors, which encompass their cryptographic keys, operational capabilities, and any additional optional details. The process involves the collection of router descriptors by directory authorities, who further generate a signed representation of the network. The aforementioned perspective is disseminated to additional directory authorities through the transmission of a summary. Later, all summaries are subjected to a voting process in order to generate a signed document, referred to as the consensus document, which provides an account of the present state of the Tor network.

In order to establish a circuit, clients acquire a roster of existing relays and then engage in a sequential exchange of session symmetric keys with each OR in the circuit. This process is commonly known as telescoping path-build design [14]. The validity of these symmetric keys is limited to the duration of the session. Perfect forward secrecy is achieved by the operation of an oblivious transfer protocol, whereby the circuit is closed and the keys are discarded by the ORs. Tor has the ability to operate with a wide range of TCP applications by utilizing the SOCKS interface, thereby eliminating the need for any modifications or kernel requirements.

2.2 Website Fingerprinting

Website fingerprinting refers to a class of traffic analysis attacks that aim to discern the specific websites (particularly front page of the website) being accessed by a user, thereby compromising their privacy. The significance of these attacks remains prominent even when

employing privacy-enhancing techniques such as anonymity networks, as they depend on metadata derived from the traces rather than decrypting the encrypted traffic. A passive adversary, present within a local network, engages in the act of monitoring direction, timing, and size of network traffic in order to infer the specific websites being accessed by user.

The threat model pertaining to the attack can be described as straightforward, involving two distinct phases: training and testing. During the training phase, the adversary engages in website visits and traffic monitoring in order to train a machine learning/deep learning model with the objective of discerning between various websites. Subsequently, the trained model is employed for the purpose of determining the specific websites that the user is accessing. Furthermore, website fingerprinting attacks can be executed in two distinct settings, namely the open-world setting and the closed-world setting. In the context of a closed setting, the adversary operates under the assumption that the user will access one of the websites being monitored. The model is then trained to discern which specific monitored website the user is visiting. In contrast, the open-world setting offers a more realistic environment where users have the ability to access any website on the Internet. However, the model is designed to discern whether a given website belongs to the monitored category or not, and if it does, to identify it.

In the subsequent discussion, we will address the primary methods employed in website fingerprinting attacks, as well as the corresponding defensive measures implemented to counteract such attacks.

2.2.1 Website Fingerprinting Attacks

Website fingerprinting attacks can be categorized broadly into two groups: those that use manually crafted features and those that use automated feature extraction. In the first type of attack, information such as the number of incoming packets, the number of outgoing packets, burst information, etc. is extracted from network traces and then fed to a machine learning classifier in order to differentiate between different websites. In contrast, in the second scenario, we leverage a deep neural network to automatically generate features from raw timing and direction data of each packet from the network trace. In the following sections, we will discuss the main attacks in each category.

Attacks using manually crafted features. Manual feature extraction attacks extract timing, direction, and size features from a network trace. These features can concentrate on a single packet of the trace, such as the timing of each packets, or on a combination of packets, such as the total number of outgoing packets. These characteristics are then used

as input for machine learning models, resulting in the development of a classifier that can differentiate between distinct websites.

Two early attacks by Herrmann et al. [23] and Cai et al [7] pave the way for cutting-edge attacks that we will later describe in this section. Herrmann et al. [23] compare observed encrypted traffic patterns with a pre-existing library of traffic fingerprints. The uniqueness of these fingerprints arises from the distinctive characteristics of web content, including HTML pages, scripts, style sheets, images, and other media objects, in terms of their structure and size.The fundamental aspect of the aforementioned technique is not contingent upon the specific sizes of the files being transmitted. However, the emphasis is placed on the patterns that are observed within encrypted IP packets. Previous methodologies relied on the dimensions of individual transmitted files, whereas the present approach is grounded in the broader patterns discernible within encrypted IP traffic. This paper presents a novel approach to fingerprinting utilising a Multinomial Naïve-Bayes classifier. The aforementioned classifier is employed for the purpose of comparing and aligning observed traffic patterns with the established repository of website fingerprints. The paper additionally assesses the effectiveness of different privacy-enhancing technologies in mitigating the aforementioned fingerprinting attack.

In their later study, Cai et al. [7] present a novel approach for conducting attacks by transforming packet traces generated by web browsers into strings. The authors propose utilising the Damerau-Levenshtein distance metric to compare these strings, taking into account the order of packets and potential disruptions in the network.

The aforementioned attacks pave the way for the use of advanced machine learning models in attacks, and options such as feature importance provide a clearer picture of which features leak the most information.

k-NN. The study conducted by Wang et al. [63] analyses packet lengths, order, and timing to discern distinctive patterns that enable the identification of visited web pages. The authors present a novel approach that utilises a k-Nearest Neighbour classifier to enhance attack methods. This approach involves the utilisation of an extensive feature set, which is further enhanced through weight adjustment techniques. This attack has been strategically designed to identify vulnerabilities in current defensive measures. Wang et al. additionally examine a complex open-world scenario, showcasing the enhanced efficacy of the new attack in comparison to prior methodologies.

CUMUL. Pancheko et al. [40] introduce the CUMUL attack. The novel methodology presented in this study abstracts the loading process of a webpage by creating a cumulative behavioural representation of its network trace. In this particular representation, relevant attributes are derived for a classification model, effectively capturing inherent traffic char-

acteristics such as packet sequencing or burst patterns. The classifier has been specifically engineered to exhibit robustness in the face of fluctuations in bandwidth, congestion levels, and webpage load timing. In order to authenticate the methodology, a comprehensive dataset comprising more than 300,000 webpages was compiled, thereby providing a more authentic portrayal of Internet traffic.

k-FP. The attack referred to as k-fingerprinting [20] utilises a modified version of random forests, which is an ensemble machine learning technique. By studying feature importance, they are the first to evaluate how and which features play a larger role in fingerprinting. The findings of the study indicate that basic characteristics, such as the number of packets, provide more substantial insights into the identification of a web page compared to intricate characteristics like packet sequencing. The effectiveness of the attack is further exemplified in a scenario where the user has unrestricted access to various online environments. This is supported by conducting tests on a total of 101,130 distinct websites, which showcases the attack's resilience when applied repeatedly and its precision.

Attacks using automated feature extraction. Automated feature extraction attacks outperform manual feature extraction attacks by utilizing deep learning techniques to represent traces in a latent feature space that is learned by the deep neural network. Input for these attacks is the feature representation of a network trace, which is either a directional vector or a directional timing vector. The direction vector is a one-dimensional representation of the direction of each packet in a network trace (+1 for outgoing packets and -1 for incoming packets). In contrast, the directional timing vector contains both timing and direction information for each packet; it is the element-wise product of timing and direction for each packet in network trace. To demonstrate the use of deep learning in website fingerprinting attacks, one of the initial papers used Stacked Denoising Autoencoder (SDAE) [1], a deep-learning technique, in conjunction with a directional vector. Tik-Tok attack [47] elucidated the significance of timing characteristics, so research shifted from using directional vector to directional timing vector.

AWF. Rimmer et al. [50] investigated three deep learning models, namely the Stacked Denoising Autoencoder (SDAE), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) in order to determine whether and by how much deep learning models outperform machine learning models. The efficacy of the deep learning-based attack is demonstrated to be more resilient in the face of alterations to dynamic web content.

When analyzing Tor traffic, Wang and Goldberg [64] show that the accuracy of website fingerprinting attacks can be enhanced by generating trace representations based on Tor cell exchanges, as opposed to the exchange of TCP/IP packets; the former providing a more consistent representation of Tor's traffic basic units. The AWF attack was the first

to use the directional vector representation of a trace as input. The direction vector is now commonly used as input for deep learning-based attacks. The direction vector represents the direction of each cell in a trace, with a value of +1 indicating outgoing cells and -1 indicating incoming cells.

DF. Sirinam et al. [55] present the concept of Deep Fingerprinting (DF), which refers to a website fingerprinting attack on Tor that leverages Convolutional Neural Networks (CNN). The attack extends the AWF attack; it demonstrates that using sophisticated CNN architecture with the same directional vector produces superior outcomes.

p1-fp. Oh et al. [37] evaluate the efficacy of Multilayer Perceptrons (MLP) and Convolutional Neural Networks (CNN) across various WF tasks, including multi-class open-world classification. Convolutional neural networks (CNNs) consistently demonstrate superior performance, even when pitted against website fingerprinting defences. The paper also demonstrates how using unsupervised deep neural networks (autoencoders) as feature extractors, can improve the performance of manual feature attacks, such as kNN or kFP.

Tik-Tok. The Tik-Tok attack [47] commonly target bursts which refer to a series of consecutive packets that are transmitted in the same direction. Previous research overlooks the significance of packet timing as a potential source of information. This study presents novel methodologies that exploit burst-level timing characteristics in order to enhance the effectiveness of website fingerprinting attacks (effectively using the same neural architecture of Deep Fingerprinting). This is accomplished by generating a directional timing vector, which is the result of performing an element-wise multiplication between the timing and direction of each packet in the trace. The results of the study indicate that the effectiveness of WF attacks is significantly improved when timing is considered in conjunction with packet direction, particularly in open-world scenarios. This implies that developers of WF defence systems should take into account burst-level timing as a noteworthy attribute that can be used for identification purposes.

Var-CNN. Bhat et al. [3] presents Var-CNN which integrates both manual and automated feature extraction techniques. The architecture of Var-CNN is derived from ResNets [21], which are convolutional neural networks commonly employed in the field of computer vision. The proposed methodology encompasses three primary principles: 1) the utilisation of dilated causal convolutions to effectively handle the distinctive arrangement of packet sequences without incurring additional computational burden, 2) the integration of manually extracted cumulative features with the deep learning model during the training process, and 3) the exploitation of packet timing information, which has been insufficiently utilised in prior website fingerprinting attacks.

All of the aforementioned techniques demonstrate high classification accuracy under conditions where the availability of training data is not a limiting factor. However, it is important to acknowledge that this scenario may pose challenges in situations where adversaries have limited access to data. The subsequent set of attacks focuses on how to conduct the most effective website fingerprinting attack with limited data.

Triplet Fingerprinting. Sirinam et al. [56] present the concept of Triplet Fingerprinting (TF), a new method for conducting website fingerprinting attacks on the Tor anonymity system. In contrast to conventional attacks that necessitate extensive training data, triplet fingerprinting utilises transfer learning over triplet networks for N-shot learning (NSL), thereby enabling it to attain a high level of accuracy with a limited number of examples per website. The TF attack has been specifically developed to enhance realism by effectively tackling obstacles such as fluctuating testing conditions (like training and testing in different location, temporal proximity of the training and testing traces) and the utilisation of outdated datasets. In particular, even when the model is trained with a dataset that is three years old, it consistently achieves an accuracy rate of approximately 85%. This level of accuracy is achieved by utilising only five examples per class. The methodology also investigates the concept of transfer learning, although its effectiveness is observed to be subpar compared to that of triplet networks.

GANDaLF. Panchenko et al. [36] present GANDaLF, an innovative methodology for Website Fingerprinting (WF) that leverages Generative Adversarial Networks (GANs). GANDaLF employs a semi-supervised learning methodology utilising Generative Adversarial Networks (GANs), which is well-suited for situations with limited data availability. The proposed approach involves training a generator model to generate synthetic traffic traces, while simultaneously training a discriminator model to distinguish between real and synthetic traces. This approach improves the performance of the discriminator, thereby enabling the effectiveness of WF even in scenarios with limited data.

AdaptiveWF. Wang et al. [62] present the concept of Adaptive Fingerprinting (AF). The utilisation of transfer learning, more specifically adversarial domain adaptation, is employed by AF in order to leverage knowledge obtained from a source dataset of significant scale and apply it to a comparatively smaller target dataset. The proposed methodology entails the utilisation of a domain adversarial network to construct a feature extractor through the implementation of a minimax game involving the feature extractor and a domain discriminator. These entities are characterised as deep neural networks. After undergoing training, the feature extractor is integrated with conventional machine learning classifiers, such as the k-nearest neighbour algorithm, in order to classify the target dataset. The findings indicate that AF has the capability to attain a high accuracy, even when working with a limited amount of data, specifically 20 traces per monitored website.

2.2.2 Website Fingerprinting Defences

The primary objective of defences against website fingerprinting is to impede the adversary's capacity to effectively execute website fingerprinting attacks. This is achieved by obscuring the genuine attributes of a website access trace through the introduction of dummy packets into the network or by implementing packet delay mechanisms. Defences are classified into different categories according to the type of defence, which will be discussed in the following sections.

Constant-rate padding. In their study, Cai et al. [5] examine the Congestion-Sensitive BuFLO (CS-BuFLO) protocol as a countermeasure against website fingerprinting attacks. CS-BuFLO extends the BuFLO [8] scheme, incorporating congestion sensitivity and rate adaptation, and is designed to be TCP-friendly and pad streams uniformly. The authors of this study suggest utilising offline-collected data to optimise the parameters of BuFLO. Additionally, they propose implementing dynamic transmission rate adaptation and enhanced stream padding techniques to reduce bandwidth usage while simultaneously increasing the level of information concealment regarding the loaded website. The primary objective of the protocol is to achieve a equilibrium between performance and security through the implementation of constraints on the rate and accuracy of adaptation.

Tamaraw [6] is an enhanced and refined version of the BuFLO protocol, specifically developed to obfuscate prominent traffic characteristics that may be vulnerable to fingerprinting attacks. The defence mechanism guarantees that in the event of two websites generating identical network traffic observations, they will remain indiscernible. Tamaraw has been specifically engineered to obfuscate critical traffic characteristics, such as the aggregate count of packets transmitted in the downstream direction. The effectiveness of the defence is assessed by comparing it to an ideal attacker, which serves as a standard for evaluation. Despite its success in preventing WF attacks, the defence incurs significant bandwidth and latency overheads, which hinder their widespread adoption in Tor.

DynaFlow [30] is a countermeasure for website fingerprinting (WF) that utilizes a constant-flow approach, which is further improved by its dynamic and customizable characteristics. The technique employs predetermined burst patterns characterized by fluctuating intervals between packets in order to obfuscate the specific website being accessed by a user. One notable characteristic of DynaFlow is its capacity for tunability, enabling it to be modified to achieve two distinct objectives. Firstly, it can be calibrated to effectively diminish attacker accuracy. Alternatively, it can be configured to reduce operational costs to a range of 30-50% while still maintaining a satisfactory level of security. The aforementioned adaptability is lacking in previous defence mechanisms that operate on a constant-flow basis, leading to consistently high levels of overhead.

The RegulaTor system [24] prioritises the implementation of traffic defences that promote regularity. Specifically, it aims to standardise the dimensions and configuration of packet bursts commonly observed in download traffic. When a sudden increase in traffic is detected, the RegulaTor system initiates the transmission of packets at a predetermined initial rate, which is subsequently reduced in accordance with a specified decay rate. In the absence of real packets, dummy packets are transmitted. The RegulaTor algorithm utilises the observed correlation between upload and download traffic in web browsing to determine the rate at which upload packets are sent. This is achieved by sending upload packets based on the download traffic rate. In contrast to alternative defence mechanisms, approach employed by RegulaTor exhibits a temporal sensitivity, prioritises the transmission of standardised bursts, and distinguishes between upload and download traffic.

Supersequence. The principle underlying Glove [35] is the process of clustering web pages into distinct groups according to their degree of similarity. By employing this approach, a mere quantity of cover traffic is sufficient to render all pages within a cluster imperceptible to potential adversaries. When a user utilises the Glove system to access a webpage, adversaries are limited to identifying the cluster to which the page belongs, but are unable to precisely determine the specific page within said cluster. The Glove system is comprised of two distinct phases: an initial offline training phase and a subsequent online defending phase. During the training phase, Glove collects web page traces, applies clustering algorithms using network features, and generates a transcript containing information about packet sizes and timings for each cluster. The aforementioned transcript is played back during the defensive phase when a user accesses page within the specified cluster.

An other common supersequencing defence is Walkie-Talkie (WT) defence. The defence strategy employed by WT [65] is centred on two primary elements, namely half-duplex communication and simulating loading of two pages by loading the supersequence of two burst sequences. Half-duplex communication facilitates the generation of succinct burst sequences that can be readily modified, allowing for the emulation of non-sensitive web pages by adding fake cells to simulate loading of two pages with minimal additional resources. These components are responsible for ensuring that timing, length, direction, and ordering of packet sequences for both sensitive pages and benign pages are made identical.

Adaptive and randomized padding. The utilisation of link padding serves to obscure traffic patterns through the introduction of deliberate delays and the inclusion of dummy messages. Nevertheless, numerous defence mechanisms impose substantial latency and bandwidth overheads, rendering them inappropriate for implementation in the Tor network. The present study presents a novel defensive mechanism known as Website Traf-

fic Fingerprinting Protection with Adaptive Defence (WTF-PAD) [26]. Derived from the concept of Adaptive Padding, WTF-PAD seeks to address the issue of Website Fingerprinting (WF) in the Tor network by providing efficient safeguarding measures that incur minimal delays and require only a moderate increase in bandwidth usage. The proposed approach utilizes receive histograms to generate padding messages as a means of responding to incoming messages, thereby simulating the behaviour of HTTP request-response interactions and altering burst patterns. Furthermore, the utilization of control messages allows the Pluggable Transport (PT) client to exert influence over the padding employed by the PT server, thereby granting the client complete authority over the padding scheme. The present system is capable of initiating transmissions and incorporates a soft stopping condition to obviate the requirement for a fixed mechanism to conceal the duration of transmission, thereby providing a competitive advantage over current defensive measures.

The current countermeasures against WF have not been widely implemented due to their significant data overheads, excessive packet delays, difficulties in implementation, or lack of effectiveness against sophisticated attacks. FRONT and GLUE [15] are two novel zero-delay lightweight defence mechanisms. The FRONT system hinders the attacker's training procedure by obscuring the feature-rich front section of traces through the utilisation of randomised dummy packets. In contrast, the GLUE system employs the technique of inserting dummy packets amidst traces, thereby creating the illusion of sequential webpage visits. This approach poses a formidable challenge to attackers, as it introduces a complex task of splitting the traces.

Traffic splitting. Henri et. al introduces a defence technique leveraging multihoming [22], where a client is connected to the Internet via multiple networks. The utilisation of a multihomed approach, in conjunction with multipath solutions such as MPTCP, facilitates the division of packets across multiple networks. This effectively increases the difficulty for an adversary who is monitoring a single network to successfully execute a wiretapping attack. The main innovation of this study is the development of a multipath scheduler known as HyWF. This scheduler has been specifically designed to distribute traffic across multiple networks in order to improve resilience against website fingerprinting, while minimising any additional traffic overhead. They additionally introduces variations of HyWF that incorporate other defensive mechanisms, namely HyWF-AP (featuring adaptive padding) and HyWF-WT (employing Walkie-Talkie). The integration of these collective defensive measures serves to enhance the level of privacy.

TrafficSliver [13] provides two lightweight defence mechanisms that rely on the concept of traffic splitting across multiple entry onion relays (ORs) in the Tor network. Instead of employing padding or implementing delays on user traffic, TrafficSliver employs a strategy of distributing user traffic across multiple discrete entry onion routers (ORs). The initial

defence mechanism functions at the network layer and employs multipathing techniques within the Tor network. The second defence mechanism involves a client-side application-layer approach, wherein either separate HTTP requests are sent for various web objects using different Tor entry ORs, or different Tor paths are utilised to request different segments of a single web object. The primary objective of the defences is to restrict the amount of data that can be observed by individual entry nodes and to disrupt traffic patterns that are susceptible to attacks by WFP.

Apart from the above mentioned classes defences can be classified as application-layer defences, adversarial traces and learning-based trace generation. Panchenko et al. [41] presented a browser plug-in that was designed to load random websites with the purpose of concealing the traffic pattern of a specific site. The HTTPOS framework [31] is designed to modify HTTP requests and manipulate the TCP behaviour in order to alter the size and timing of packets and/or web objects. LLaMA [11] is a client-side mechanism that introduces random delays to outgoing HTTP requests, while also injecting dummy HTTP requests. ALPaCA [11] is a server-side defence mechanism that employs the insertion of dummy web objects, or the padding of existing ones, in order to standardize the size of various websites. The Mockingbird system [46] produces traces that exhibit resistance against white-box attacks, specifically targeting an adversary with the capability to train a classifier using previously defended traces. The Dolos framework [54] is capable of disrupting deep learning classifiers used in the field of wireless communication by generating input-agnostic adversarial patches. These patches are designed to guide the injection of dummy packets into traffic traces. The technique known as BLANKET [34], aims to counter deep learning website fingerprinting attacks by perturbing the features of live connections without prior knowledge. Surakav et al. [17] investigate the application of generative adversarial networks (GANs) for the purpose of emulating authentic traffic patterns exhibited by various webpages.

2.3 Website Fingerprinting in Next-Gen Networks

Mobile and satellite networks are examples of next-generation networks that assist in solving Internet connectivity issues in remote areas. This section examines prior investigations into website fingerprinting in the context of next-generation networks.

2.3.1 Mobile Networks

LTE, also referred to as Long-Term Evolution, represents a standardized framework established by the 3rd Generation Partnership Project (3GPP) consortium [2]. Its primary objective is to enhance data transmission rates and connectivity beyond the capabilities of preceding 3G networks. The device functions across various frequency ranges and employs sophisticated modulation methods. The Fifth Generation (5G) technology represents an advancement over Long-Term Evolution (LTE) [19] by offering enhanced data rates, reduced latency, and the ability to support a significantly larger number of interconnected devices concurrently. The system utilizes novel spectrum bands, employs Massive MIMO (Multiple Input Multiple Output) technology [29], and implements network slicing to accommodate a wide range of applications, including smartphones, IoT devices, and autonomous vehicles.

Mobile networks and fiber-based networks exhibit distinct characteristics. Mobile networks often have higher latency due to the time it takes for signals to travel through the air and the processing time at various network nodes. They can experience congestion more readily, as many users share the same bandwidth. The performance can be inconsistent, with speeds varying depending on the user's proximity to a cell tower, the number of active users, and the type of technology. WF attacks on mobile networks were also shown to achieve good accuracy. Rupprecht et al. [28] provide an analysis of the vulnerabilities and potential attacks that can be targeted towards LTE (Long-Term Evolution), a prevalent mobile communication standard. LTE, despite its superior transmission capabilities and robust security measures, is vulnerable to a range of attacks, such as denial-of-service attacks, downgrade attacks, and attacks targeting identification and localization. There is a significant emphasis placed on the layer-two, also known as the data link layer, of the LTE protocol stack, which has received relatively less attention in terms of security investigation. In their study, Rupprecht et al. present a series of attacks targeting this particular layer, including an identity mapping technique that allows for website fingerprinting on encrypted LTE traffic.

The authors conduct a series of experiments and case studies to show feasibility of website fingerpriting, which provided insights into the application of advanced classification techniques to LTE like kNN and SVM using features extracted from layer two by virtue of radio network temporary identifier, packet data convergence protocol and downlink control information metadata. These techniques demonstrate notable success rates in accurately identifying both sites and users. The results emphasise the pressing necessity to tackle these vulnerabilities, particularly in light of the upcoming implementation of

5G technologies, which might possess comparable protocol specifications and, as a result, similar vulnerabilities.

2.3.2 Satellite Networks

Satellite networks are comprised of a fusion of ground stations and orbiting satellites, enabling long-range communication, typically spanning worldwide extents. These networks play a vital role in regions where ground communication infrastructures, such as cables or towers, are not feasible to be deployed. Satellite communication operates through the transmission of signals from a ground station to a satellite, which subsequently relays the signal to either a receiving ground station or directly to user terminals. The utilization of this method of communication holds significant importance across a range of applications, encompassing global broadcasting, navigation, and Internet connectivity [9].

Low Earth Orbit (LEO) and Geostationary Orbit (GEO) satellites are two primary types of satellites used in these networks. Low Earth Orbit satellites are positioned at varying altitudes, typically ranging from around 180 km to 2,000 km above the Earth's surface. The close proximity of these entities to Earth provides the advantage of reduced communication delays, leading to decreased latency and increased potential for enhanced data throughput [60]. Nonetheless, the coverage area of these systems is relatively limited, thereby requiring the deployment of extensive constellations in order to achieve global coverage. In contrast, geostationary satellites are situated at an altitude of approximately 35,780 km above the Earth's surface, maintaining a synchronous orbit with the Earth's rotational movement. This characteristic gives the impression of immobility in relation to a stationary reference point on the Earth's surface, enabling them to traverse extensive regions, frequently encompassing entire continents. The increased distance between entities results in a corresponding increase in latency during communication. GEO satellites are frequently employed for the purpose of broadcast services, whereas emerging LEO constellations have the objective of offering broadband Internet services with decreased latency. When employing satellites in geostationary orbit, the Earth's distance forces a round-trip time greater than 550 milliseconds [43]. Combined with the limited and shared capacity of the physical link, this poses a challenge to the quality of traditional Internet access.

Ma et. al [32] offer a comprehensive examination of the operational effectiveness of LEO satellite networks, with specific emphasis on the Starlink network developed by SpaceX. Starlink, as the most extensive LEO satellite network constellation at present, provides a user-friendly service that allows end-users to easily connect to the Internet. The primary objective of Starlink is to offer Internet connectivity that is on par with traditional terrestrial networks. The study raised inquiries regarding the performance of Starlink, the

factors that exert influence on its operations, and its extent of global coverage. The study yielded several significant findings:

The throughput and latency of Starlink exhibit a high degree of dynamism, presenting notable distinctions when compared to conventional terrestrial networks. It is a common occurrence for users to encounter service disruptions. The performance also varies by geography: authors observe 2.3× higher delay in the USA, compared to the UK, as well as 2.6× lower throughput (on average). They also found instances of unusually high packet loss of up to 50%, with over 12% of samples obtaining more than 5% packet loss [27]. The performance and power consumption of Starlink is significantly influenced by various environmental factors such as terrain, solar storms, precipitation, cloud cover, and temperature [27]. Authors observed 2× increase in median Page Transit Time for the same web services when accessed on a day with moderate rain, as compared to a clear sky day.

It is worth noting that Starlink manages to sustain a steady flow of data despite transitions; nevertheless, as previously stated, there are performance concerns that do not exist in fiber-based networks. The aforementioned concerns distinguish this network type from fiber and LTE networks; the extent to which this distinction contributes to website fingerprinting has yet to be investigated.

Summary

Effective and efficient website fingerprinting attacks exist for terrestrial networks including fiber and LTE, and say that website fingerprinting attacks are under-explored in the satellite connectivity setting. The satellite communication links of popular satellite constellations like Starlink are substantially different from what one can find in fiber, there is a question of how successful website fingerprinting attacks can be on this setting. Another important concern is on the deployability of defences: can they offer the same security guarantees with similar overheads, or would these be substantially different? The aforementioned observations offer substantial evidence of the disparities in network characteristics between satellite and terrestrial networks. The efficacy of network analysis attacks, such as website fingerprinting attacks, in satellite-based networks is a subject of inquiry.

Chapter 3

Methodology

In this section, we detail the methodology of our study towards ascertaining the susceptibility of LEO satellite links to website fingerprinting attacks. In Section 3.1, we detail the goals of our study and the approach we follow in our evaluation. Section 3.2 describes the experimental testbed we used for crawling websites via Tor, and Section 3.3 details the process we followed for collecting and pre-processing the obtained data. Lastly, in Section 3.4, we characterize the collected data, highlighting the differences between fiber and Starlink collected traffic considering a set high-level traffic characteristics.

3.1 Study Goals and Approach

The study conducted in the scope of this thesis has two main goals: (i) to shed light over the potential vulnerability of Tor users making use of satellite Internet connections against website fingerprinting attacks, and; (ii) to assess whether existing website fingerprinting defences can efficiently safeguard Tor user's privacy over such connections. To this end, our study will consider an adversary that can eavesdrop the connection between a Tor user and her intended destination, and apply sophisticated website fingerprinting attacks towards identifying which website the user might be visiting over a satellite Internet link. Later in our study, we will assume the possibility for Tor users to make use of website fingerprinting defences, towards making the task of the adversary more difficult. We are then interested in evaluating whether the usage of existing website fingerprinting defences remain their efficiency and effectiveness when applied to satellite Internet links instead of typical fiber links.

Next, in Section 3.1.1, we describe the threat model and the assumptions we considered for developing our study. Then, in Section 3.1.2, we describe the website fingerprinting attacks and defences we consider to be available to the adversary and, respectively, to Tor users. Lastly, Section 3.1.3 describes the metrics of interest for our evaluation of the success of website fingerprinting attacks and defences.

3.1.1 Assumptions and Threat Model

We follow the typical threat model for website fingerprinting attacks, albeit with one important change on the location and mode of operation of the adversary. While the typical website fingerprinting adversary is usually co-located with a user and can eavesdrop on their connection towards launching a website fingerprinting attack, this is not possible to be accomplished in the same way when considering a Starlink satellite link. The main reason being that Starlink satellites beam data using sophisticated signal encryption schemes that allow only the target satellite dish receiver to be able to decode the information being sent/received to/from the satellite [67]. In other words, even if the adversary places a Starlink satellite dish within the same geographical data transmission cell where the target user satellite dish sits in, they would not be able to access the raw IP packet stream that is directed at the target user. This prevents a typical website fingerprinting adversary from inspecting user data.

In other cases, such as website fingerprinting attacks launched over LTE/4G networks, the adversary is first required to tap into the radio signals exchanged between a user's equipment (e.g., a smartphone) and the LTE base station. This capability, which can be obtained through the use of LTE software stacks implemented in software-defined radios in tandem with sniffer analysis frameworks, allows the adversary to access and decode transmissions ranging from the physical layer up to the data-link layer, and then derive user-specific traffic metadata. However, to the best of our knowledge, such capabilities are not publicly available for Starlink satellite links, despite current advances in the reverse-engineering of Starlink downlink signals. For this reason, we introduce a variation of the website fingerprinting adversary model, which we describe next.

An ISP-based website fingerprinting adversary. While strong signal encryption may prevent third parties from inspecting the traffic of satellite Internet users, website fingerprinting attacks might still be useful for the ISPs operating the satellite networking service itself. In this setting, it is possible that, despite allowing users to leverage privacy-preserving protocols such as those used in Tor, snooping ISPs might wish to identify which content is being accessed by their users, e.g., towards preventing the access to websites

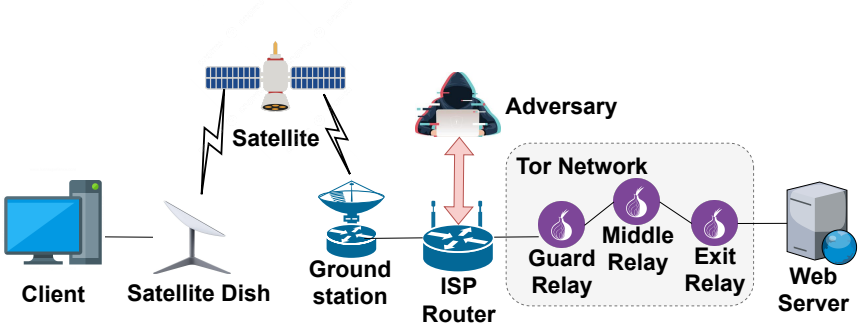


Figure 3.1: Threat model.

that allow for the streaming of DRM-protected content [51]. This setting, illustrated in Figure 3.1, places the eavesdropping adversary at the satellite provider's IXP (Internet eXchange Point), with full access to the IP traffic exchanged by the satellite Internet user and their destination. Given that the ISP we assume in our threat model is responsible for managing satellite and fiber links, we also assume it possesses the ability to differentiate between the kind of links different users might make use of when connecting to the Internet via its network infrastructure.

We gather network access traces directly from the client's machine, i.e., the only infrastructure component under our control. In future work, we will explore the deployment of a proxy node that will mediate and record the traffic exchanged between the client and the Tor network, thus assuming a more natural position in the above threat model. Examining the communication performance / privacy trade-offs involved into placing an additional node on the path for conducting our measurements is also left for future work.

Other assumptions. Our study operates under the assumption of a closed-world setting, where the adversary assumes that the client is exclusively visiting one of the monitored websites. This configuration intentionally gives more power to the adversary, allowing it to build traffic classification models specific to a predefined pool of websites. We assume that the access to any monitored website in our closed-world setting is equally probable. We also assume that the attacker is able to separate the traces associated with the loading of different websites and determine which defence is in use by a Tor user. Thus, the use of website fingerprinting defences like GLUE [15] is outside the scope of our study.

3.1.2 Attacks and Defences

In this section, we describe the website fingerprinting attacks and defences considered throughout our study.

Website fingerprinting attacks. In our study, we employ some of the most prominent attacks in the literature, including the k-FP, deep fingerprinting (DF), and Tik-Tok attacks.

k-FP is one of the most effective website fingerprinting attacks based on manual feature engineering (Section 2.2). The attack algorithm extracts 175 features from the trace through an examination of the quanta's size, direction, and timing. Before employing a k-Nearest Neighbors classifier to predict website accesses, the classifier constructs a unique fingerprint for each site through a modification of the Random Forest algorithm.

As previously covered in Section 2.2, DF and Tik-Tok are based on deep neural architectures using convolutional neural networks, which directly extract latent features from input traces passed to the classifier during the training and inference step. The DF attack accepts as input a direction vector representing the direction of n packets in the trace. The Tik-Tok model also requires a trace formatted in a vector-shape, but which is composed of directional-timing information, i.e., the input vector is the element-wise product of direction and timing of n packets in a trace. We do not alter the architectural models used in the DF or Tik-Tok attacks nor their default hyper-parameters, with the exception of the "patience" hyperparameter, which we increase from 3 to 6. In order to prevent overfitting during the training of deep learning models, early stopping employs a "patience" parameter as a form of regularization. By increasing the patience parameter in early stopping from 3 to 6, the training process would conclude after six epochs without any improvement in validation loss. This is advantageous in cases where the model's validation loss encounters brief plateaus or fluctuations during training.

Website fingerprinting defences. To avoid the repeated collection of traffic traces for evaluating website fingerprinting defences, these defences' authors oftentimes release defence simulators that can turn undefended Tor traffic traces into their defended versions in an offline manner. While such a process could in principle raise some suspicions about the faithfulness of simulations, Gong et al. [16] have recently compared the simulation and true implementation results for a set of WF defences and reached the conclusion that simulators can accurately reflect the application of defences on live traffic.

Given the above, we utilize the defence simulators and configurations recently used in the work of Veicht et al. [61], which focused on the security analysis of website fingerprinting defences. We make use of open-source implementations of the WTF-PAD [26], FRONT [15], CS-BuFLO [5], and Tamaraw [6] defences.

Specifically, we have incorporated WTF-PAD [26] into our setup, utilizing the implementation provided in the WFES [10] repository. Additionally, we have integrated two versions of FRONT [15] into our experiments, named FRONT_T1 and FRONT_T2. FRONT_T1 is configured with parameters $N_c = N_s = 1700$, $W_{min} = 1$, and $W_{max} = 14$, whereas FRONT_T2 uses different settings, with $N_c = N_s = 2500$. Due to the larger sampling window in FRONT_T2, it is expected to introduce more dummy packets into the trace than FRONT_T1. For CS-BuFLO [5] and Tamaraw [6], we have employed a set of other specific parameters. CS-BuFLO is set with $d = 1$ and a range for $2^{-4} * 1000 \leq \rho \leq 2^3 * 1000$. Tamaraw, on the other hand, utilizes $\rho_{out} = 0.04$, $\rho_{in} = 0.012$ with $L = 50$.

3.1.3 Evaluation Procedure and Metrics

Evaluation procedure. During our evaluation, we make use of 10-fold cross-validation when training and testing our classifiers to minimize the effects of selection bias. In particular, we employ stratified cross-validation to ensure an equal distribution of instances across all the classes comprising our dataset. In each cross-validation fold, we use 80% of the data for training, 10% for the model's validation, and the remaining 10% for testing.

Attack performance metrics. The main metric we pay attention to when analyzing the success of a website fingerprinting attack (whether a defence is being used or not) is *accuracy*. Accuracy has been extensively used in the website fingerprinting literature for determining the efficacy of both attacks and defences in the closed-world scenario, carrying a rather intuitive meaning for an adversary – it quantifies the adversary's success in discerning which website a given user is accessing. More specifically, accuracy is defined as the ratio of correctly predicted instances to the total number of instances in the dataset.

Defence performance metrics. Apart from a desirable reduction in attacks' accuracy, website fingerprinting defences may also be evaluated on the amount of overhead they impose over an undefended Tor network trace. For this reason, in our experiments, we also leverage additional *bandwidth* and *latency* as the critical efficiency indicators of website fingerprinting defences. Defences are typically deemed to be practical if and only if they can substantially reduce an attack's accuracy while having a small impact on latency and bandwidth overheads.

Traffic analysis machine. To train and test our models on the network traces we collected for our study, we leverage a server machine with 2x AMD EPYC 7302 16-Core CPU, 512 GB RAM, and an NVIDIA A100 GPU w/40 GB memory.

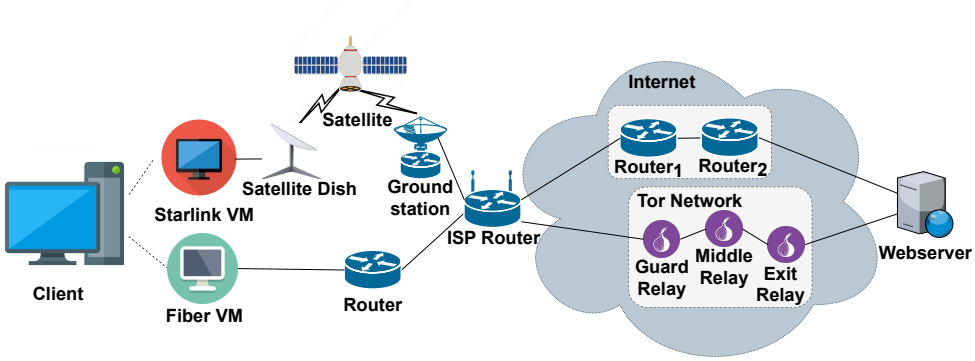


Figure 3.2: Data collection process overview.

3.2 Experimental Testbed

Figure 3.2 depicts a birds-eye overview of our experimental testbed. It essentially comprises a client machine under our control, which will be used for connecting our client to the Tor network towards accessing a set of websites included in our closed-world website list.

The client machine employed in our testbed comprises a physical machine that executes two virtual machines (VMs), each with 60GB storage and 2GB of RAM. The machine is equipped with two network interface cards, and each VM routes its traffic through a different interface. One of these cards is connected to a Starlink dish (satellite Internet connection), while the other interface is attached to our university's fiber-based network (terrestrial Internet connection). On each VM, we deploy Docker containers that we orchestrate for simultaneously collecting network traces of a given website through the fiber and Starlink link.

Our setup's main advantage is that it can help us deal with concept drift [25], which can have a significant impact on website fingerprinting experiments. Specifically, the ability to collect traces for the same website simultaneously over the two different links enables us to collect traces that represent a given webpage in roughly the same instant of time, thus minimizing the chance that our fingerprint database would include significantly different versions of a given webpage, should, for instance, all traffic via fiber be collected after all traffic collected via Starlink. In the next section, we describe our data collection and pre-processing procedure in detail.

3.3 Data Collection and Pre-Processing

We aim to collect a novel dataset of website accesses that contain two different shares of traces: those collected when the client uses a simple terrestrial fiber network, and those collected when the client uses a satellite link. Having both of these shares allows us to build a baseline of the effectiveness of website fingerprinting attacks on a typical Internet connection, thus allowing for direct comparisons with the effectiveness of the same attacks once deployed over data collected via the Starlink connection.

Websites included in our dataset. In our data collection procedure, we considered a closed-world of websites composed of the top 125 websites found on the Tranco list [45]. We manually verified that each of these websites are active, by sending a request to the frontpage of the website and confirming we would get back an HTTP 200 response code. The front page of a website may include pictures, scripts, trackers, and elements such as navigation menus, hyperlinks, multimedia (like videos and audio), forms, and interactive features. The full list of websites included in our dataset can be found in Listing A.1.

Furthermore, we collect a total of 125 instances of each of the 125 websites, albeit the number of valid samples is reduced due to transmission errors detected after data pre-processing (discussed in the next few paragraphs).

Collection of Tor traces. We perform the visits to each website using `tbselenium`, a headless wrapper around the Tor browser. We used the default configuration setup for Tor and, to ensure the freshness of each website visit via Tor, we restarted the `tbselenium` Tor driver after clearing its cache upon each visit.

We divide our data collection in batches; in each batch, we collect trace information for a single website when it is accessed via Tor, and then we repeat this process for all 125 websites. In the event that a given batch includes a request that returns an explicit error to `tbselenium` (e.g., due to network instabilities), we revisit the website (up to a maximum of 3 visits) until we receive a valid response.

To coordinate the experiment, the client VMs running the docker containers were synchronized through an orchestrating daemon running in the main OS, using the Python Flask REST API [18]. The daemon sends requests to both machines simultaneously, and manages the placement of requests via Tor. It also takes into account any error or timeout response received during website crawling and act accordingly (typically by ordering new requests while deleting erroneous requests).

Collection of plain Firefox traces. Apart from the above Tor-focused dataset, we also collect additional website traces over direct connections to each website using Firefox. The

main reason why we collect these traces is to study and characterize the overheads of using Tor instead of Firefox on both satellite and fiber connections. We also use this data to compare the performance of a classifier when fingerprinting plain traffic vs. Tor traffic.

As before, we collect traces using the fiber and Starlink connections simultaneously. We also interleave Tor and Firefox requests on our data collection procedure, by including each fiber/Starlink Firefox-based request in each batch of requests sent via Tor. More concretely, each batch is deemed as valid if and only if we successfully access a website via Tor (on both interfaces) and the same website via Firefox (also on both interfaces).

Pre-processing of network traces and dataset configurations. After collecting our traces, we undergo a pre-processing phase where we aim to weed-out from the dataset those traces that resulted in timeouts or that include errors such as blank pages, captcha pages and blocked pages (possibly because of censorship based on exit nodes in Tor circuits [68]) that prevent the website from being fetched correctly (but that did not trigger explicit errors in `tbselenium`). We consider a request to timeout if one minute has elapsed before the page can be successfully retrieved.

After removing traces afflicted by the above issues, and uniformizing the number of valid samples per website, we obtained a dataset which final configuration includes 80 instances of 75 different websites visited over both Starlink and terrestrial fiber, using both Tor and Firefox (which we collectively denote as $TorFirefox\text{-}SatFiber_{4\times75\times80}$), and containing a total of $4 \times 75 \times 80 = 24\,000$ samples. This makes each of the four shares of the $TorFirefox\text{-}SatFiber_{4\times75\times80}$ dataset comparable to the DS-19 dataset [15], which includes 100 websites with 100 instances each, and which has been used for the benchmarking of website fingerprinting attacks and defences.

Changing the representation of network traces. Our data collection process yields raw packet capture files (`.pcap`) which include a vast amount of data that is not required for our analysis. To ease the process of feature extraction for our classifiers, we start by generating an intermediate representation of the traces which can be efficiently analyzed and that includes inter-packet timing, direction, and size of every packet in a trace. Similarly to earlier works, we represent packet direction as either +1 for outgoing packets or -1 for incoming packets. Depending on the classifier under test, we extract the necessary information from this representation to generate features (Section 2.2.1): summary statistics (for k-FP), directional vectors (for DF), or directional-timing vectors (for Tik-Tok).

3.4 Characterization of Starlink and Fiber Traces

In this section, we study the traces comprising our datasets in order to uncover what major differences in performance exist between Internet connections established between terrestrial fiber and Starlink, as well as highlighting the performance drops expected when using Tor instead of Firefox in these different networking environments. We now present a characterization of the traces included in the $TorFirefox\text{-}SatFiber_{4\times75\times80}$ dataset, and describe our main takeaways below.

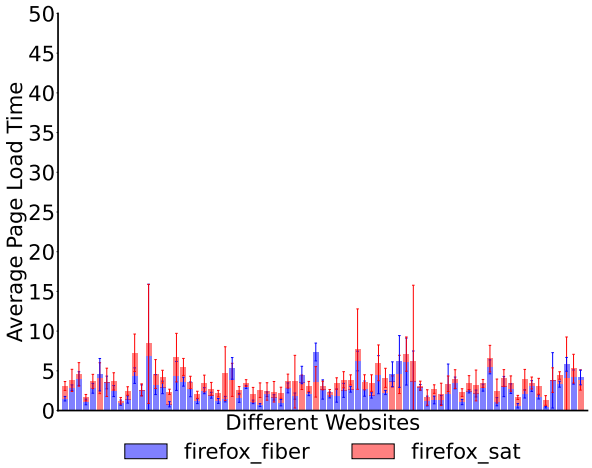


Figure 3.3: Avg. page load time (Firefox).

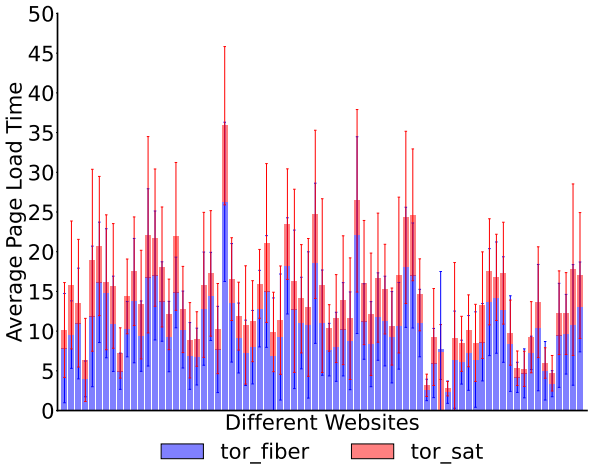


Figure 3.4: Avg. page load time (Tor).

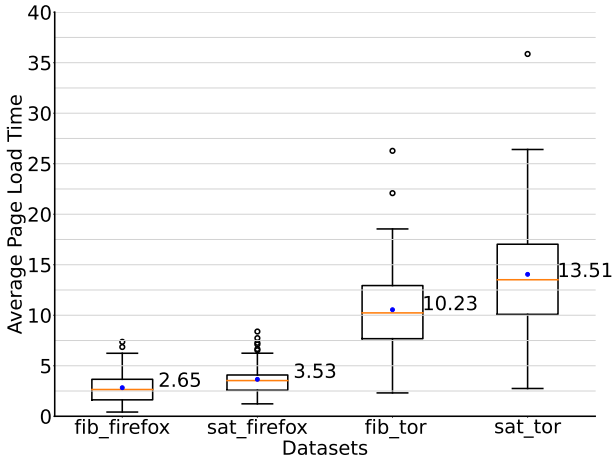


Figure 3.5: Box plot of avg. page load times.

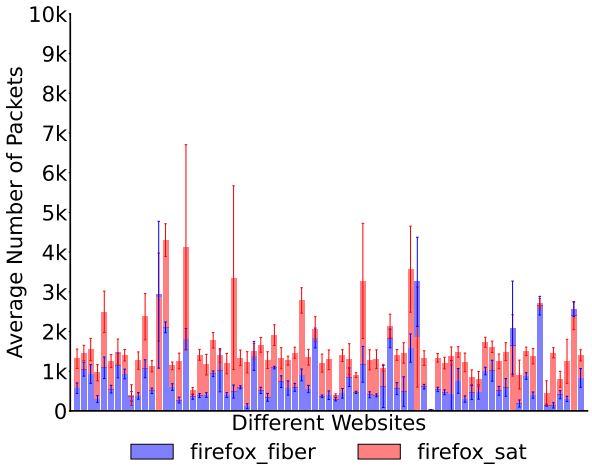


Figure 3.6: Avg. no. of packets (Firefox).

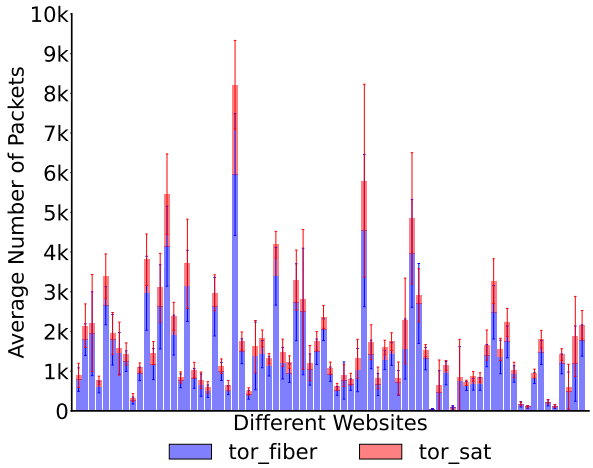


Figure 3.7: Avg. no. of packets (Tor).

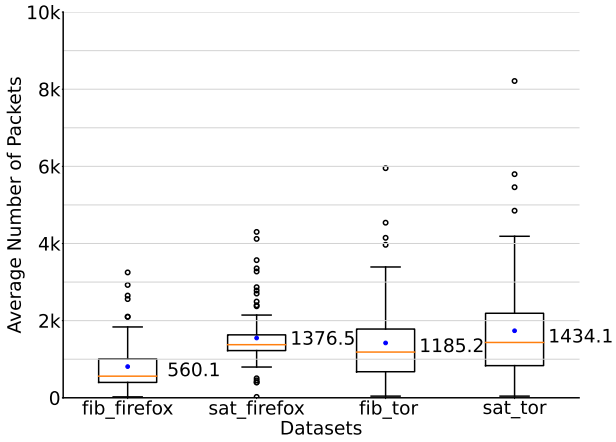


Figure 3.8: Box plot of avg. no. of packets.

Tor over Starlink is 3.83 times slower than Firefox over Starlink. Figure 3.3 and Figure 3.4 depict the average page load time observed when loading each of the 75 websites, over the traces collected via Firefox and Tor, respectively. From the box plot in Figure 3.5, we can more easily see that Starlink-based connections consistently reveal higher times-to-last-byte when compared to fiber connections, representing a total increase of the median of 33.2% when considering website accesses established over Firefox, and 32% average increase when considering website accesses performed via Tor.

The largest difference observed, however, is on the use of Tor when compared to the use of Firefox. For fiber connections, the median of Tor accesses is 3.86× slower than those

via Firefox. This difference is evident even when considering Starlink connections, where the median of Tor accesses is 3.83× slower than those via Firefox.

Starlink connections require a larger exchange of network packets. Figure 3.6 and Figure 3.7 depict the average number of packets observed when accessing each of the 75 websites via Firefox and Tor, respectively. In addition, the box plot in Figure 3.8 allows us to perform an easier comparison between the trends observed in each share of the $TorFirefox\text{-}SatFiber_{4\times75\times80}$ dataset. The median number of packets exchanged when using Firefox more than doubles when using a Starlink connection (1376.47 packets) when compared to the use of fiber (560.11 packets). While for Tor we are also able to observe an increase in exchanged packets when moving from a fiber to Starlink setting, this increase is less pronounced (∼21% more packets).

Interestingly, website accesses using Firefox via Starlink reveal a smaller inter-quartile range than accesses via a terrestrial fiber connection, indicating a more concentrated distribution around the mean. The opposite is true for accesses over Tor where, albeit less pronounced, the distribution seems to be more concentrated around the mean for the connections making use of the fiber connection.

TCP retransmission requests are more common in Starlink traffic. Towards understanding the differences in the number of packets observed in our traces (Starlink vs. fiber), we conducted an additional analysis focused on the study of TCP retransmissions. Figure 3.9 and Figure 3.10 display a per-website breakdown of the average percent of packets retransmitted observed across our traces, while Figure 3.11 summarizes this information. In general, we can observe that retransmission requests are rather rare throughout our traces, but more common in Starlink data exchanges.

For instance, when accessing the website *googledomains.com* via Firefox, we observed that 0.02% of packets are retransmitted when using the fiber setting, while 0.3% of packets are retransmitted when using the Starlink setting. When accessing the same website using Tor, we find that 0.03% of packets are retransmitted over a fiber connection, while 0.7% of packets are retransmitted over Starlink. This indicates that even if the percentage of retransmitted packets is not excessively high, there is a notable disparity in the percentage of packet retransmissions when utilizing the Starlink network configuration compared to the fiber setting. We hypothesize that this may be caused by the inherent noise that is characteristic of satellite Internet connections.

Starlink-exchanged packets tend to be smaller than fiber-exchanged ones. Figure 3.12 and Figure 3.13 depict the average length of packets observed when accessing each of the 75 websites via Firefox and Tor, respectively. The overall trend of packet lengths can be more easily grasped by looking at Figure 3.14. We can see from the figure that

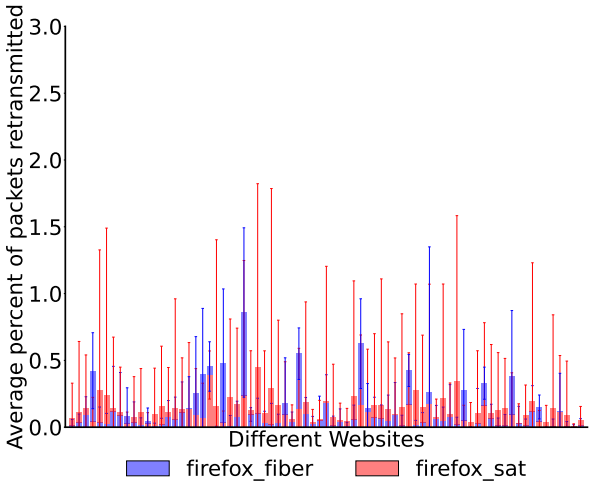


Figure 3.9: Avg. retransmissions (Firefox).

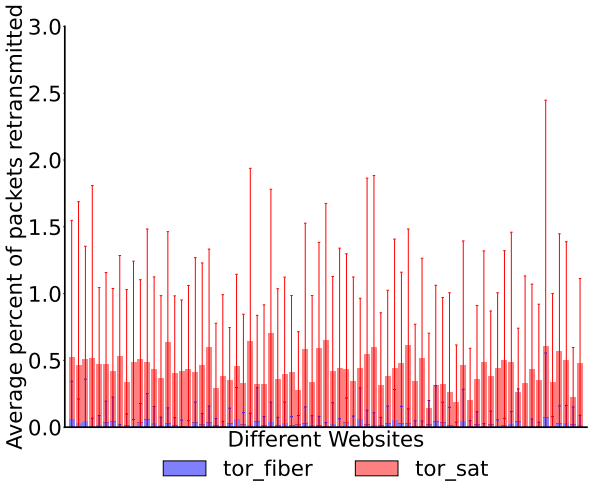


Figure 3.10: Avg. retransmissions (Tor).

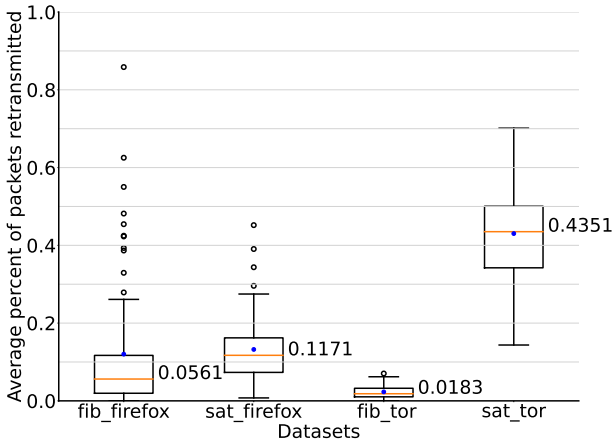


Figure 3.11: Avg. percent of retransmissions.

the packets composing Tor traffic exhibit a rather concentrated size, with a median size of 1501.79 when Tor data is exchanged via Starlink and a median size of 1785.11 when exchanged via fiber connections. Interestingly, we observe that the size of Firefox packets is also rather concentrated around a mean of 1471.64, while Firefox packets exchanged over fiber connections exhibit a more variable (and typically larger) length, with a median of 2188.94 and a size of 3254.31 at the 75th percentile. Satellite connections such as Starlink exhibit increased latency and potentially elevated error rates as a result of the extensive distances that signals must traverse. In order to address these problems, the Starlink connection could employ smaller packets, which can be retransmitted more rapidly in the

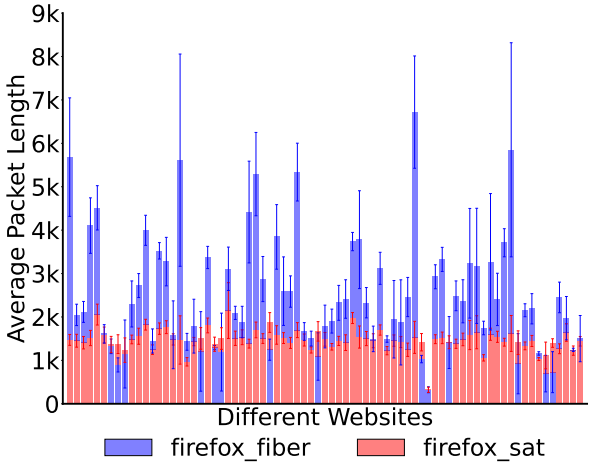


Figure 3.12: Avg. packet length (Firefox).

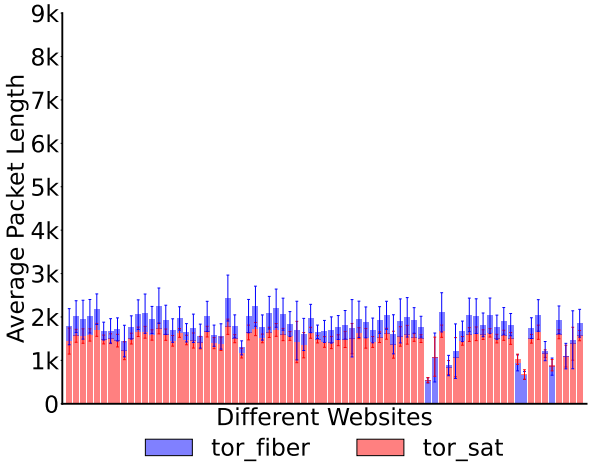


Figure 3.13: Avg. packet length (Tor.)

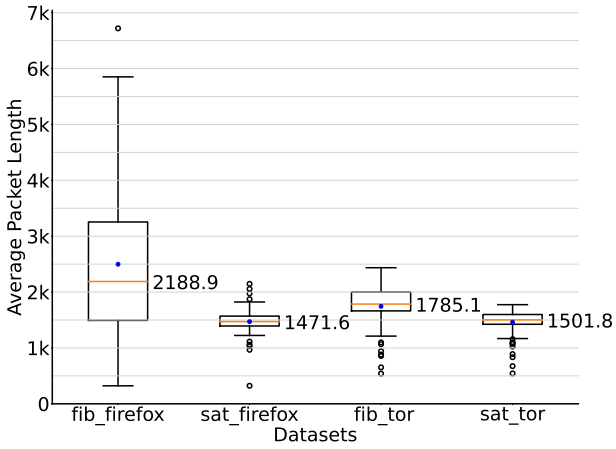


Figure 3.14: Box plot of avg. packet length.

event of errors. This approach also decreases the waiting time for packet acknowledgment, thereby enhancing overall efficiency.

Tor exchanges a comparable number of cells on Starlink and fiber connections. Figure 3.15 displays a per-website breakdown of the mean number of Tor cells observed across our traces, while Figure 3.16 summarizes this information. In contrast to the average number of packets exchanged (Figure 3.8), the average number of Tor cells transmitted through both fiber and Starlink exhibit a expected similarity. According to the data presented in Figure 3.16, one can observe that 75% of the traces exhibit a number of Tor cells that is less than or equal to 5436. This is an interesting observation, as most deep-learning website fingerprinting attacks trim their input vectors to 5000 cells. These results

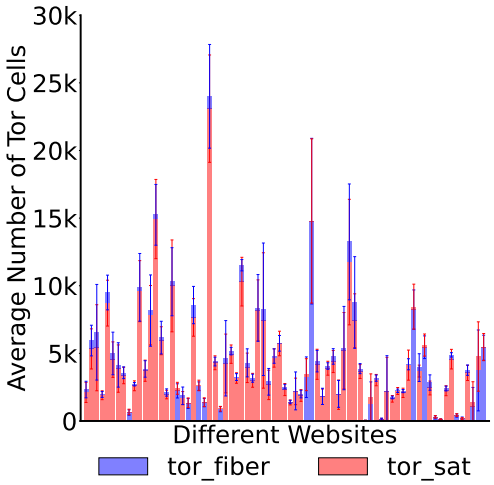


Figure 3.15: Avg. no. of Tor cells.

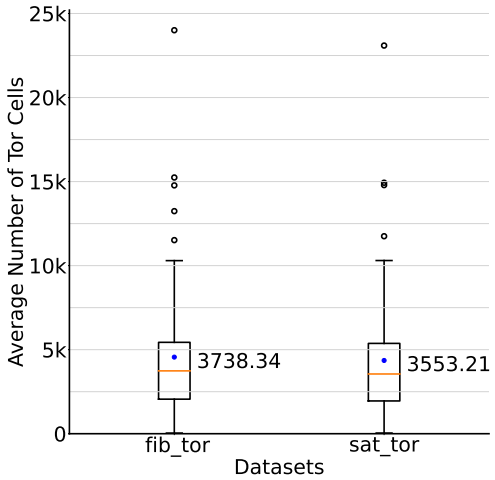


Figure 3.16: Box plot of avg. no. of cells.

show that a trimming threshold close to this value is also expected to work well for our dataset, concurring with the results of our experiments in Chapter 4. The distribution of the average number of Tor cells is also quite similar for traces collected using both fiber and Starlink (with similar inter-quartile range). Each is positively skewed as the median is closer to the lower whisker and the number of outliers is comparable.

Summary. Overall, our findings suggest that the use of satellite links impose a larger relative penalty on plain Firefox connections rather than Tor connections. We hypothesize that the variable latency and jitter introduced and compounded by the multiple relays composing Tor circuits may help amortize the performance penalties incurred by clients that use satellite uplinks to connect to the Internet via Tor. In the following chapter, we discuss our findings resulting from the experiments conducted over our datasets leveraging state-of-the-art website fingerprinting attacks and defences.

Chapter 4

Evaluation

Upon obtaining necessary data for the analysis, it is evident that the average number of packets/tor cells and average page load time differ between fiber and Starlink. This chapter examines in greater detail how Starlink compares to fiber in terms of effectiveness against website fingerprinting attacks and defences. We provide an analysis of the outcomes of website fingerprinting attacks on Starlink and fiber networks in Section 4.1. A comparison of the effectiveness and overheads of defences against website fingerprinting attacks on Starlink networks as opposed to fiber networks is the subject of Section 4.2.

4.1 Website Fingerprinting on Starlink Connections

In this section, we compare the susceptibility of fiber and Starlink connections to different website fingerprinting attacks. Section 4.1.1 is dedicated to evaluating website fingerprinting attacks using manual feature extraction, while Section 4.1.2 is focused on examining website fingerprinting attacks using deep learning techniques.

4.1.1 Attacks with Manually Engineered Features

k-FP is one of the most effective manual feature-engineering-based website fingerprinting attacks. The algorithm derives 175 features by analyzing the size, direction, and timing of packets in the trace. In addition, these features are rated according to their impact on the attacks. We start by providing details on the accuracy of k-FP on the different sets of traces composing our dataset. Besides assessing the success of this attack on Tor

Dataset	k-FP	k-FP (w/ pkt. lengths)	DF	Tik-Tok
Firefox - Fiber	0.8557	0.8892	0.8837	0.7945
Firefox - Starlink	0.4075	0.4285	0.4915	0.4715

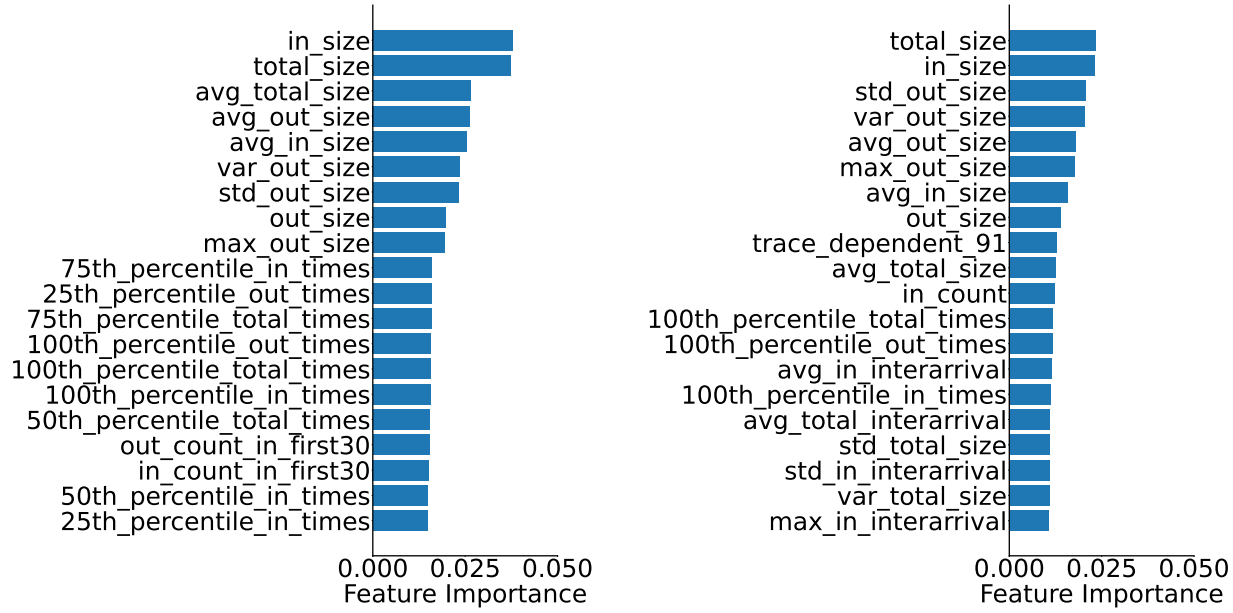
Table 4.1: Attack accuracy for Firefox traces (on TCP/IP data).

traffic, we also attempt to fingerprint plain Firefox connections. Note that, in practice, the destination of Firefox connections would be trivially disclosed to an adversary (e.g., by looking at the connection's destination IP address). However, we do this as an exercise towards understanding how satellite connections affect traffic features and whether these effects degrade or improve our ability to fingerprint network traffic.

To perform the above comparisons with Firefox traffic, we modify the original implementation of the k-FP attack in two important ways. First, we allow for features to be directly generated from TCP/IP header information (e.g., IP packet length, time between IP packets, etc.) instead of Tor cells as in the original attack. Second, we create a version of the k-FP classifier which takes packet lengths into account as features for building and matching website fingerprints. The rationale for these modifications on k-FP hinges on the fact that Tor exchanges data in Tor cells padded to 512B, thus making packet length analysis irrelevant [14]. In contrast, Firefox does not exchange data via cells, thus providing an analyst with access to raw TCP/IP packet length information.

k-FP is more accurate for Firefox traffic over fiber. In this first experiment, we used raw TCP/IP packet header data (direction, size, and timing) to generate features for the k-FP attack. Table 4.1 depicts the accuracy of the website fingerprinting attacks we consider on Firefox, over both fiber and Starlink connections. We see that the original k-FP classifier achieves an accuracy of 85% when fingerprinting websites accessed via Firefox - Fiber, but achieves an accuracy of only 40% when fingerprinting websites accessed via Firefox - Starlink. The inclusion of packet lengths in k-FP attack brings only marginal benefits for the attack in both settings, amounting to an accuracy increase of $\sim$3%.

Figure 4.1a and Figure 4.1b show the top-20 most important features for the k-FP attack when launched over Firefox traffic exchanged via fiber and Starlink, respectively (Hayes et al. discuss the description of features used in k-FP attack [20]). We can observe that the two most important features for classifying website accesses on Firefox via fiber is the sum of all incoming packet sizes and the sum of all packet sizes in the data exchange, whereas the importance of these features is exchanged for Firefox via Starlink traffic. We can also see from Figure 4.1a that 9 out of the 20 most important features focus on packet timing information, whereas only 7 timing related features are within the top 20 most important features for Firefox traffic exchanged over Starlink. Interestingly, while timing features



(a) Via fiber.

(b) Via Starlink.

Figure 4.1: Top-20 most important features (Firefox traces).

in the former case are mostly related to percentiles, timing features are more related to the average and standard deviation of packet arrivals for the latter. The 25th percentile of timings for all outgoing packets, which is the second most important feature in k-FP attack on Firefox over fiber, drops to the eleventh position when k-FP with packets length is used; the feature that is most significant for the former (50th percentile of interarrival timings of incoming packets) is not included in the list of the top 20 features for the latter. It is also interesting to note that the top features of Firefox over fiber leak more data than the top features of Firefox over Starlink. We can conclude, based on all the information regarding feature importance analysis, that the significance of features in Firefox traces over both fiber and Starlink settings is as follows: direction, size, and finally timing. The significantly higher number of retranmissions (more than two times) in Starlink traces compared to fiber traces can contribute to increased noise in the traffic pattern. This, in turn, may be a crucial factor in explaining the inferior performance of the kFP attack on Starlink traces.

k-FP is more accurate for Tor traffic over fiber. In this second experiment, we analyze the effectiveness of the original k-FP attack on Tor traffic exchanged via fiber and Starlink. Thus, in this case, we extract the attack features based on our estimates of the

Dataset	k-FP	DF	Tik-Tok
Tor - Fiber	0.7282	0.8738	0.8860
Tor - Starlink	0.6426	0.8540	0.8682

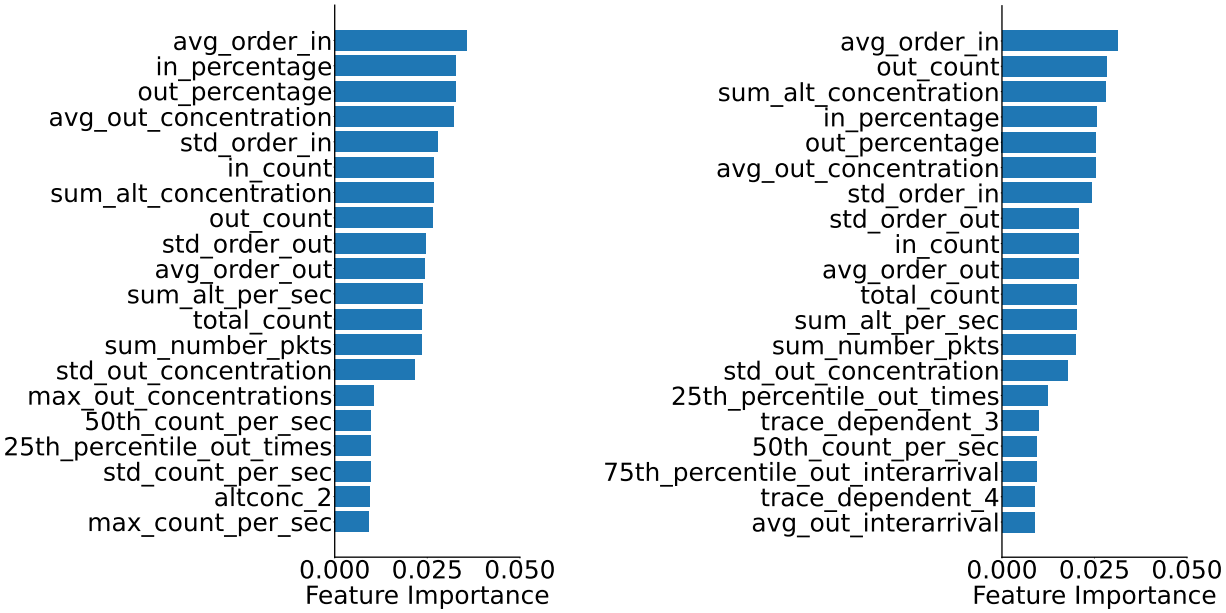
Table 4.2: Attack accuracy using Tor cell data.

Tor cells exchanged within these traces. Tor uses Transport Layer Security (TLS) records to encrypt its data. After parsing the packets in the trace, an attacker can reconstruct TLS records in their entirety. Since Tor cells are the fundamental unit of traffic (each Tor cell is 512 bytes in length), each TLS record includes an integral number of Tor cells. Consider the following example: the trace contains TLS records with the following lengths: -544, 1088, -1088, -544, 1088. Now the first TLS record in the sequence will contain one Tor cell, the second will contain two, and so on. Therefore, the directional vector that most accurately represents the Tor traffic will be -1, 1, 1, -1, -1, -1, -1, 1, 1. We generate the Tor cell information for all Tor traces and use that for evaluation. The results in Table 4.2 show that the accuracy of k-FP in Tor - Fiber traffic is close to 73% and around 64% for Tor - Starlink. This discrepancy is consistent with the results observed for Firefox browsing, where the classifier had performed better for fingerprinting websites visited via the fiber connection.

A close look at the top-20 most important features for classifying Tor traffic (Figure 4.2a and Figure 4.2b) reveals that the cumulative average of incoming packets is the most important feature for classifying both kinds of connections. Moreover, 14 out of the top-20 features are shared between both (though not necessarily in the same order). This may be the case due to the similarity observed in Tor cell statistics for both fiber and Starlink traffic, as observed in Figure 3.16. Nevertheless, the remaining features in the top-20 exhibit some variations (e.g., the inclusion of "trace dependent" features in Starlink traffic) which may be explained by noise that is inherent to satellite connections.

4.1.2 Attacks with Deep Learning

Deep fingerprinting and Tik-Tok fingerprinting are automated feature extraction attacks that use deep learning models for classification. Tik-Tok fingerprinting accepts a directional timing vector as input, whereas deep fingerprinting accepts a directional vector. In order to execute deep learning classifiers, the vector input length must be consistent across all instances. Prior research employs a length of 5000 [50] [55] [47] as the input vector length; therefore, instances with lengths below 5000 are appended with zeros until they reach



(a) Via fiber.

(b) Via Starlink.

Figure 4.2: Top-20 most important features (Tor traces).

5000, and those with lengths greater than 5000 are truncated to match 5000. This section outlines the key findings of our experiments.

The success of deep learning attacks is comparable to k-FP on Firefox traffic. The results in Table 4.1 show that the DF and Tik-Tok deep learning attacks achieve a comparable accuracy to classical machine learning attacks like k-FP when fingerprinting Firefox traffic. Specifically, we see that the accuracy of DF is comparable to the accuracy obtained by k-FP when considering packet sizes. These results suggest that Firefox traffic is easily fingerprintable by less sophisticated classifiers, and that the application of deep learning attacks brings only marginal improvements, if any – for instance, Tik-Tok achieves an accuracy of only 79%, which is around 10% below the accuracy obtained by k-FP without considering packet size information. The average number of packets for Firefox traces over fiber is 560, whereas for Firefox over Starlink, it is 1376 (Figure 3.8). In order to ensure proper execution of deep learning, it is necessary to establish the input length of the directional vector at 5000 for both DF and Tik-Tok fingerprinting attacks. To ensure that the average number of packets reaches 5000, zeros are added to the directional vector until it reaches a length of 5000. Consequently, the majority of data stored in the directional

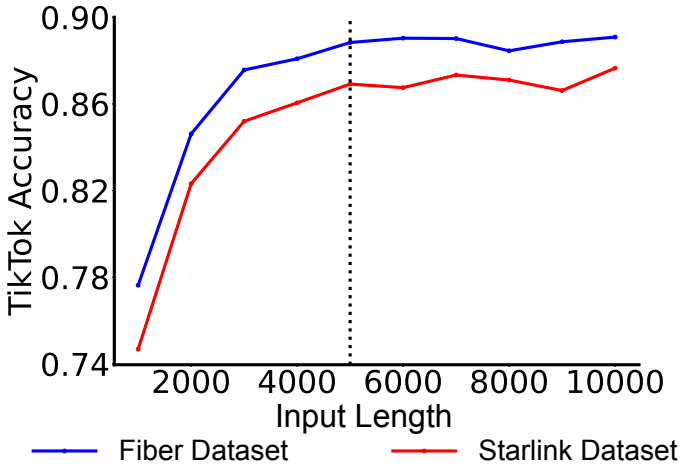


Figure 4.3: Trade-off between trace length and Tik-Tok accuracy.

vectors for Firefox consists of zeros, which poses a challenge for deep learning attacks to surpass the effectiveness of manual feature extraction attacks.

Deep learning attacks can successfully fingerprint Tor traffic via fiber and Starlink. The accuracy results reported in Table 4.2 reveal that the DF and Tik-Tok attacks achieve a similar performance when applied to Tor traffic regardless of whether the traces were collected via fiber or Starlink connections. Interestingly, we can also observe that the DF classifier is able to achieve roughly the same accuracy for Firefox traffic collected over fiber (Table 4.1) and Tor traffic, suggesting that users have little benefits when using Tor for shielding their web browsing behaviors.

Impact of trace length on fingerprinting accuracy. As mentioned in Section 3.1.2, DF and Tik-Tok automatically extract latent features from a trace's direction or direction+timing representation, respectively. In the original attacks, the number of cells considered in each trace (n) is set at 5000 (instances with lengths below 5000 are appended with zeros, and those with lengths greater than 5000 are truncated). While our previous analysis in Section 3.4 – Figure 3.16 suggested that the number of cells in a trace was relatively similar, irrespective of whether a Tor trace had been collected via fiber or Starlink, we also observed that $n = 5000$ roughly corresponded to the 75th percentile of cells across all traces.

To ascertain whether this value of n is adequate for the current version of Tor and across the two networking environments under test, we devised an experiment where we vary n and assess the impact of this choice on the accuracy obtained by the DF classifier.

Figure 4.3 depicts the outcome of this experiment. We can observe that the evolution of accuracy according to n is rather similar for both our Tor fiber and Starlink traces. The figure also shows that an n smaller than 4000 prevents the classifier from achieving an accuracy over 86% for both connection types, and that $n = 5000$ offers a good trade-off between accuracy and the length of input traces.

Training Data	**Testing Data**	**Accuracy**
Fiber	Fiber	0.8860
Fiber	Starlink	0.8168
Starlink	Fiber	0.8627
Starlink	Starlink	0.8682

Table 4.3: Tik-Tok's accuracy for different training and testing datasets.

Models trained on Starlink dataset are more robust. Table 4.3 presents the accuracy of the Tik-Tok attack when alternating the shares of the dataset used for training and testing the classifier. We can see that using Tor traces collected on the fiber connection to train an attack that aims to fingerprint Tor traffic exchanged via Starlink results in an accuracy decrease of about 4% when compared to the use of Starlink training data. In turn, using Tor traces collected on the Starlink connection to train an attack that aims to fingerprint Tor traffic exchanged via fiber results in an accuracy decrease of only 2% when compared to the use of fiber training data. The above results suggest that an adversary who trains the Tik-Tok attack on traces obtained via Starlink can obtain a relatively high accuracy when fingerprinting both Starlink and fiber traffic. A potential explanation for this fact is that Starlink connections possess an inherent noise, contributing to an increased per-class sample diversity and an overall enhancement of the model's robustness.

4.2 Defending Starlink Connections against WF

This section presents a comparative analysis of the effectiveness and efficiency of fiber and Starlink connections in relation to various website fingerprinting defences. Section 4.2.1 is specifically devoted to assessing the effectiveness of different website fingerprinting defences on both types of connections, while Section 4.2.2 primarily examines the efficiency of these defences.

Defences	Fiber Traces	Starlink Traces
WTF-PAD	0.8360	0.7880
FRONT_T1	0.5910	0.4700
FRONT_T2	0.5462	0.4358
CS-BuFLO	0.1655	0.1540
Tamaraw	0.1087	0.1008
No defence	0.8860	0.8682

Table 4.4: Tik-Tok accuracy against Tor with different defences.

4.2.1 Defences's Effectiveness

As mentioned in Section 3.1.2, our defence evaluation makes use of a set of defence simulators that convert undefended Tor cell traces into defended traces. We leverage the simulators to generated defended versions of the Tor traces obtained via fiber and Starlink connections, and assess the effectiveness of defences when applied to both networking environments. This section presents the main takeways of our experiments.

Defences are more effective on the Starlink dataset. Table 4.4 lists the accuracy of the Tik-Tok attack on defended Tor traffic, both for Starlink and fiber-collected traces. Overall, we can observe that the accuracy obtained for Starlink traces is reduced when compared to the accuracy observed for fiber traces. While this observation was also true for non-defended traffic (see Table 4.2), the application of constant-rate defences like Tamaraw and CS-BuFLO achieve similar accuracy reductions, bringing the attack's accuracy down to approximately 10% and 16%, respectively. This is expected, as both defences heavily shape the timing and sizes of packets sent to the network in order to obfuscate traffic patterns.

While other defences can also moderately decrease the Tik-Tok attack's accuracy, we can observe that the application of these defences result in a disparate effectiveness when applied to fiber and Starlink traces. For instance, we can see that the FRONT_T1 and FRONT_T2 defence variants reduce the attack's accuracy for an extra 12% and 11% when deployed on Starlink traces. While less pronounced, this trend can also be observed for the WTF-PAD defence, where its application to Starlink traces leads to an accuracy reduction of about 5%. FRONT defences introduce random padding to the front of the trace, which, when combined with the inherent noise in the satellite network, enhances the effectiveness of defences on the Starlink dataset. These results suggest that the incorporation of dummy traffic, although generally effective on fiber, has a comparatively greater impact

Dataset	Tamaraw	WTF-PAD	CS-BuFLO	FRONT_T1	FRONT_T2
Fiber	5.83	1.00	30.54	1.00	1.00
Starlink	4.28	1.00	21.50	1.00	1.00

Table 4.5: Latency overhead for different defences

on the ability of traffic classifiers to accurately fingerprint Tor connections established over Starlink.

4.2.2 Defences's Overhead

After gauging the effectiveness of defences over the Tor traces included in our dataset, we now turn our attention to compare the overheads imposed by these defences when applied to fiber and Starlink traces. When reporting our results, we present the overall bandwidth and latency overhead imposed by each defence as the median value of the bandwidth usage and latency values observed amongst the defended traces.

The bandwidth function calculates the total data usage of a given network trace. It computes the total number of bytes transmitted in the trace. The latency function computes the total time elapsed between the first and the last packet in a network trace. This is done by subtracting timestamp of the first packet from that of the last packet.

Defended Starlink traces impose a smaller latency overhead. Table 4.5 shows the latency overhead of the considered defences when applied to fiber and Starlink traces. Overall, one can observe that latency overhead tends to be the same (or less pronounced) when applied to Starlink traces than when the same defence is applied to fiber traces. Note that the latency overhead is effectively zero on both kinds of traces for adaptive and random padding defences like FRONT variants and WTF-PAD, since these defences largely aim to avoid the introduction of communication delays. However, considering Tamaraw and CS-BuFLO, defended Starlink traces demand for about 1.36 and 1.42 times less latency than fiber traces, respectively.

The bandwidth overhead of defended Starlink and fiber traces is similar. Table 4.6 also shows the bandwidth overhead of the defences. For instance, it shows that Tamaraw, the most bandwidth-inefficient defence, imposes an overhead of 1.6 times that of a Tor undefended trace over fiber. We can also see from the table that Starlink traces impose an equivalent or slightly larger bandwidth overhead than that of fiber traces, for all the considered defences. This increase in overhead is particularly noticeable for the

Dataset	Tamaraw	WTF-PAD	CS-BuFLO	FRONT_T1	FRONT_T2
Fiber	1.60	1.18	1.48	1.24	1.34
Starlink	1.65	1.21	1.47	1.31	1.46

Table 4.6: Bandwidth overhead for different defences

FRONT_T2 defence, where the bandwidth overhead is about 12% larger when the defence is applied to Starlink traces rather than fiber traces.

While the tested defences allow for a reduction in attack accuracy on Starlink connections (see Table 4.4), the above analysis reveals that the application of the defences leads to a small increase in bandwidth usage, but which can nevertheless pose a concern for satellite Internet users whose satellite ISPs may apply data caps towards regulating traffic, e.g., during busy parts of the day [27].

Chapter 5

Conclusion

5.1 Concluding Remarks

This thesis examines the vulnerability of Tor users to website fingerprinting when data is transmitted through Low Earth Orbit (LEO) satellite Internet connections. In order to investigate this, we generate a unique dataset by creating experimental testbed that simultaneously gather Tor and non-Tor traffic from both Starlink and fiber connections. We empirically illustrate the distinctions in characteristics between Firefox traces and Starlink traces for both non-Tor and Tor traffic. Our findings indicate that Starlink-exchanged packets are typically smaller in size compared to fiber-exchanged packets. Additionally, we observed a higher frequency of TCP retransmission requests in Starlink traffic. Our investigation revealed that website fingerprinting attacks on non-Tor traffic are only 50% as effective on Starlink connections compared to fiber connections, due to the contrasting network characteristics. These variations in network characteristics seem to be nullified by the interference generated in the Tor circuit. Consequently, website fingerprinting attacks exhibit comparable effectiveness in both Starlink and fiber connections within the Tor traffic. Furthermore, we notice that the inherent noise in the Starlink connection generates natural adversarial examples, which serve as valuable training data for the creation of a website fingerprinting attack model that exhibits high robustness.

Additionally, we analyze the behaviour of website fingerprinting defences in satellite Internet connections. While website fingerprinting defences can reduce the effectiveness of attacks on Starlink connections compared to fiber connections, the analysis reveals that implementing these defences leads to a marginal rise in bandwidth consumption. However, this rise in usage may cause concern for satellite Internet users who experience data caps.

5.2 Limitations and Future Work

This section discusses the limitations of our study and points to directions for future work.

Tor circuit going via multiple Starlink satellites. We investigate the effect of a single Starlink connection on website fingerprinting in our experimental configuration, whereby traffic travels via Starlink before arriving at the guard node. In order to study how website fingerprinting changes when there is more than one Starlink connections between two consecutive nodes in the tor circuit, as well as how it changes when there are multiple Starlink connections across the entire tor circuit, future work would involve creating an environment with multiple satellite connections.

Geo-distributed Starlink testbed. A single node linked to Starlink was taken into account in our evaluation setup. Because of the configuration of the satellite constellation and the number of active subscribers in particular regions, recent research has revealed [27] that the performance of Starlink client nodes may differ across continents (or even countries). Further research is required to determine whether and how the results presented in this study can be generalized to other geographical areas. To achieve this, future endeavours will involve the installation of additional Starlink data collection nodes in various parts of the world.

Considering the influence of weather. While meteorological conditions do not have an effect on underwater fiber cables, they do indeed affect satellite transmissions. Our data collection was concluded under clear sky conditions. However, previous research on the performance of LEO satellites [27] has indicated that various weather conditions can have an impact on the Internet connectivity of the satellite (for instance, by introducing extra jitter and latency). An intriguing avenue for future research entails gathering website access traces across various weather conditions (e.g., snow, rain, clouds) in order to determine whether these conditions may introduce substantial variations in an adversary's capability to execute precise website fingerprinting. Additionally, we discovered that the model trained on the Starlink dataset exhibited greater robustness in comparison to the model trained on the fiber dataset. By subjecting the model to training in diverse weather conditions, we can potentially obtain a model that is even more adaptable and robust.

Lack of open-world experiments. The focus of our research was website fingerprinting in a closed-world environment where the adversary assumes the client is accessing the monitored website. Moving forward, our objective is to assess the vulnerability of Starlink traffic to website fingerprinting in an open-world environment where clients can access unmonitored sites, thereby offering a more accurate representation of internet browsing.

References

[1] Kota Abe and Shigeki Goto. Fingerprinting attack on Tor anonymity using deep learning. *Proceedings of the Asia-Paci c Advanced Network*, 42:15–20, 2016.

[2] David Astely, Erik Dahlman, Anders Furuskär, Ylva Jading, Magnus Lindström, and Stefan Parkvall. Lte: the evolution of mobile broadband. *IEEE Communications Magazine*, 47(4):44–51, 2009.

[3] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. Var-cnn: A data-efficient website fingerprinting attack based on deep learning. *Proceedings on Privacy Enhancing Technologies*, 2019(4):292–310, 2019.

[4] Vaibhav Bhosale, Ahmed Saeed, Ketan Bhardwaj, and Ada Gavrilovska. A characterization of route variability in leo satellite networks. In *Proceedings of the International Conference on Passive and Active Network Measurement*, pages 313–342. Springer, 2023.

[5] Xiang Cai, Rishab Nithyanand, and Rob Johnson. Cs-buflo: A congestion sensitive website fingerprinting defense. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 121–130, 2014.

[6] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. A systematic approach to developing and evaluating website fingerprinting defenses. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 227–238, 2014.

[7] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, pages 605–616, 2012.

[8] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 605–616, 2012.

[9] Vincent WS Chan. Optical satellite networks. *Journal of Lightwave Technology*, 21(11):2811, 2003.

[10] Giovanni Cherubin. Bayes, not naïve: Security bounds on website fingerprinting defenses. *Proceedings on Privacy Enhancing Technologies*, 4:135–151, 2017.

[11] Giovanni Cherubin, Jamie Hayes, and Marc Juárez. Website fingerprinting defenses at the application layer. *Proc. Priv. Enhancing Technol.*, 2017(2):186–203, 2017.

[12] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. Online website fingerprinting: Evaluating website fingerprinting attacks on tor in the real world. In *Proceedings of the 31st USENIX Security Symposium*, pages 753–770, 2022.

[13] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. Trafficsliver: Fighting website fingerprinting attacks with traffic splitting. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1971–1985, 2020.

[14] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, page 21, USA, 2004. USENIX Association.

[15] Jiajun Gong and Tao Wang. Zero-delay lightweight defenses against website fingerprinting. In *Proceedings of the 29th USENIX Security Symposium*, pages 717–734, 2020.

[16] Jiajun Gong, Wuqi Zhang, Charles Zhang, and Tao Wang. Wfdefproxy: Modularly implementing and empirically evaluating website fingerprinting defenses. *arXiv preprint arXiv:2111.12629*, 2021.

[17] Jiajun Gong, Wuqi Zhang, Charles Zhang, and Tao Wang. Surakav: generating realistic traces for a strong website fingerprinting defense. In *Proceeding of the 2022 IEEE Symposium on Security and Privacy*, pages 1558–1573, 2022.

[18] Miguel Grinberg. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.", 2018.

[19] A. Gupta and R. K. Jha. A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232, 2015.

[20] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *Proceedings of the 25th USENIX Security Symposium*, pages 1187–1203, 2016.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Sébastien Henri, Gines Garcia-Aviles, Pablo Serrano, Albert Banchs, and Patrick Thiran. Protecting against website fingerprinting with multihoming. *Proceedings on Privacy Enhancing Technologies*, 2020(2):89–110, 2020.

[23] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pages 31–42, 2009.

[24] James K Holland and Nicholas Hopper. Regulator: A straightforward website fingerprinting defense. *Proceedings on Privacy Enhancing Technologies*, 2022.

[25] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, page 263–274, Scottsdale, Arizona, USA, 2014.

[26] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. Toward an efficient website fingerprinting defense. In *Proceedings of the European Symposium on Research in Computer Security*, pages 27–46, 2016.

[27] Mohamed M Kassem, Aravindh Raman, Diego Perino, and Nishanth Sastry. A browser-side view of starlink connectivity. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 151–158, 2022.

[28] Katharina Kohls, David Rupprecht, Thorsten Holz, and Christina Pöpper. Lost traffic encryption: fingerprinting lte/4g traffic on layer two. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 249–260, 2019.

[29] Erik G. Larsson, Ove Edfors, Fredrik Tufvesson, and Thomas L. Marzetta. Massive mimo for next generation wireless systems. *IEEE Communications Magazine*, 52(2):186–195, 2014.

[30] David Lu, Sanjit Bhat, Albert Kwon, and Srinivas Devadas. Dynaflow: An efficient website fingerprinting defense based on dynamically-adjusting flows. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society*, pages 109–113, 2018.

[31] Xiapu Luo, Peng Zhou, Edmond W. W. Chan, Wenke Lee, Rocky K. C. Chang, and Roberto Perdisci. Httpos: Sealing information leaks with browser-side obfuscation of encrypted flows. In *Proceedings Network and Distributed System Security Symposium - Volume 11*, 2011.

[32] Sami Ma, Yi Ching Chou, Haoyuan Zhao, Long Chen, Xiaoqiang Ma, and Jiangchuan Liu. Network characteristics of leo satellite constellations: A starlink-based measurement from end users. In *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2023.

[33] François Michel, Martino Trevisan, Danilo Giordano, and Olivier Bonaventure. A First Look at Starlink Performance. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 130–136, Nice, France, October 2022.

[34] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. Defeating {DNN-Based} traffic analysis systems in {Real-Time} with blind adversarial perturbations. In *Proceeding of the 30th USENIX Security Symposium*, pages 2705–2722, 2021.

[35] Rishab Nithyanand, Xiang Cai, and Rob Johnson. Glove: A bespoke website fingerprinting defense. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 131–134, 2014.

[36] Se Eun Oh, Nate Mathews, Mohammad Saidur Rahman, Matthew K. Wright, and Nicholas Hopper. Gandalf: Gan for data-limited fingerprinting. *Proceedings on Privacy Enhancing Technologies*, 2021:305–322, 2021.

[37] Se Eun Oh, Saikrishna Sunkam, and Nicholas Hopper. p1-fp: Extraction, classification, and prediction of website fingerprints with deep learning. *Proceedings on Privacy Enhancing Technologies*, 2019(3), 2019.

[38] One Web. https://oneweb.net/. Last Accessed: 2023-12-13.

[39] Rolf Oppliger. *SSL and TLS: Theory and Practice*. Artech House, 2023.

[40] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *Proceedings of the 23rd Annual Network and Distributed System Security Symposium*, 2016.

[41] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114, 2011.

[42] James Pavur, Martin Strohmeier, Vincent Lenders, and Ivan Martinovic. Qpep: An actionable approach to secure and performant broadband from geostationary orbit. In *Proceedings 2021 Network and Distributed System Security Symposium*, Feb 2021.

[43] Daniel Perdices, Gianluca Perna, Martino Trevisan, Danilo Giordano, and Marco Mellia. When satellite is all you have: watching the internet from 550 ms. In *Proceedings of the 22nd ACM Internet Measurement Conference*, pages 137–150, 2022.

[44] Ani Petrosyan. https://www.statista.com/statistics/617136/digital-population-worldwide/. Accessed: 2023-12-13.

[45] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Proceedings of the Network and Distributed Systems Security Symposium*, 2019.

[46] Mohammad Saidur Rahman, Mohsen Imani, Nate Mathews, and Matthew Wright. Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces. *IEEE Transactions on Information Forensics and Security*, 16:1594–1609, 2020.

[47] Mohammad Saidur Rahman, Payap Sirinam, Nate Mathews, Kantha Girish Gangadhara, and Matthew Wright. Tik-tok: The utility of packet timing in website fingerprinting attacks. *Proceedings on Privacy Enhancing Technologies*, 2020(3), 2020.

[48] Aravindh Raman, Matteo Varvello, Hyunseok Chang, Nishanth Sastry, and Yasir Zaki. Dissecting the performance of satellite network operators. *Proc. ACM Netw.*, 1(CoNEXT3), nov 2023.

[49] RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3. https://www.rfc-editor.org/rfc/rfc8446. Last Accessed: 2023-12-13.

[50] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated website fingerprinting through deep learning. In *Proceedings of the Network and Distributed System Security Symposium*, February 2018.

[51] Rogers Media Inc. v. John Doe 1, 2022 FC 775 (CanLII). https://canlii.ca/t/jpncf. Last Accessed: 2023-12-13.

[52] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. Breaking lte on layer two. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 1121–1136, 2019.

[53] Raffaello Secchi, Pietro Cassarà, and Alberto Gotta. Exploring machine learning for classification of quic flows over satellite. In *ICC 2022-IEEE International Conference on Communications*, pages 4709–4714. IEEE, 2022.

[54] Shawn Shan, Arjun Nitin Bhagoji, Haitao Zheng, and Ben Y Zhao. Patch-based defenses against web fingerprinting attacks. In *Proceedings of the 14th ACM Workshop on Arti cial Intelligence and Security*, pages 97–109, 2021.

[55] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1928–1943, 2018.

[56] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1131–1148, 2019.

[57] Starlink. https://www.starlink.com/. Last Accessed: 2023-12-13.

[58] Starlink - new dish specifications. https://www.starlink.com/specifications?spec=4. Last Accessed: 2023-12-13.

[59] Qixiang Sun, Daniel R Simon, Yi-Min Wang, Wilf Russell, Venkata N Padmanabhan, and Lili Qiu. Statistical identification of encrypted web browsing traffic. In *Proceedings of the 23rd IEEE Symposium on Security and Privacy*, pages 19–30, Oakland, CA, USA, 2002.

[60] F. Vatalaro, G.E. Corazza, C. Caini, and C. Ferrarelli. Analysis of leo, meo, and geo global mobile satellite systems in the presence of interference and fading. *IEEE Journal on Selected Areas in Communications*, 13(2):291–300, 1995.

[61] Alex Veicht, Cedric Renggli, and Diogo Barradas. Deepse-wf: Unified security estimation for website fingerprinting defenses. *Proceedings on Privacy Enhancing Technologies*, 2023(2), 2023.

[62] Chenggang Wang, Jimmy Dani, Xiang Li, Xiaodong Jia, and Boyang Wang. Adaptive fingerprinting: Website fingerprinting over few encrypted traffic. In *Proceedings of the 11th ACM Conference on Data and Application Security and Privacy*, pages 149–160, 2021.

[63] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. In *Proceedings of the 23rd USENIX Security Symposium*, pages 143–157, 2014.

[64] 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.

[65] Tao Wang and Ian Goldberg. Walkie-talkie: An efficient defense against passive website fingerprinting attacks. In *Proceedings of the 26th USENIX Security Symposium*, pages 1375–1390, 2017.

[66] Yaoying Zhang, Qian Wu, Zeqi Lai, and Hewu Li. Enabling Low-latency-capable Satellite-Ground Topology for Emerging LEO Satellite Networks. In *Proceedings of the 2022 IEEE Conference on Computer Communications*, pages 1329–1338, Virtual Event, May 2022.

[67] Yu Zhang, Shuangrui Zhao, Ji He, Yuanyu Zhang, Yulong Shen, Xiaohong Jiang, et al. A survey of secure communications for satellite internet based on cryptography and physical layer security. *IET Information Security*, 2023, 2023.

[68] Zhao Zhang, Wenchao Zhou, and Micah Sherr. Bypassing tor exit blocking with exit bridge onion services. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, page 3–16, 2020.

APPENDICES

Appendix A

Websites used for experimentation

Listing A.1 includes the websites contained in the $TorFirefox\text{-}SatFiber_{4\times75\times80}$ dataset.

1. adobe.com
2. amazon.co.jp
3. amazon.in
4. apache.org
5. apple.com
6. azure.com
7. bbc.co.uk
8. bbc.com
9. bing.com
10. bit.ly
11. booking.com
12. cdc.gov
13. cnn.com
14. digicert.com
15. dnsmadeeasy.com
16. doubleclick.net
17. dropbox.com
18. ebay.com
19. etsy.com
20. facebook.com
21. fandom.com
22. fastly.net
23. fbcdn.net
24. flickr.com
25. force.com
26. gandi.net
27. github.com
28. github.io
29. google-analytics.com
30. googledomains.com
31. icloud.com
32. instagram.com
33. intuit.com
34. issuu.com
35. linode.com
36. live.com
37. mail.ru
38. microsoft.com
39. mozilla.org
40. msn.com
41. naver.com
42. netflix.com
43. nytimes.com
44. office365.com
45. opera.com
46. oracle.com
47. outlook.com
48. paypal.com
49. pornhub.com
50. reddit.com
51. reuters.com
52. salesforce.com
53. salesforceliveagent.com
54. skype.com
55. soundcloud.com
56. sourceforge.net
57. spotify.com
58. stackoverflow.com
59. t.me
60. telegram.org
61. theguardian.com
62. tiktok.com
63. tumblr.com
64. twitch.tv
65. vimeo.com
66. w3.org
67. weebly.com
68. wellsfargo.com
69. whatsapp.com
70. wikimedia.org
71. wikipedia.org
72. xvideos.com
73. yahoo.co.jp
74. youtube.com
75. zemanta.com

Listing A.1: List of websites drawn from the Tranco list considered in our experiments.