EAI Endorsed Transactions
on Security and Safety
Research Article EAI.EU

WebTracker: Real Webbrowsing Behaviors – is Website Fingerprinting no1w Realistic?

Taryn Chovan[1], Daisy Reyes[1], Eno Dynowski[1], John Mikos[1], Eric Chan-Tin[1,*], Mohammed Abuhamad[1], Shelia Kennison[2]

[1]Loyola University Chicago, USA
[2]Oklahoma State University, USA

Abstract

The increasing demand for privacy has driven the adoption of privacy-enhancing tools such as VPNs, but website fingerprinting – the analysis of packet metadata like packet size and number of packets – still poses a substantial risk. Website fingerprinting allows adversaries to predict a victim's web usage based on their browsing patterns, effectively creating a "fingerprint". Recent studies have largely focused on laboratory settings and have assumed a simplified model: a victim visits a single website at a time and that all network packets can be observed. However, a new private browser extension, WebTracker, deployed with real users, shows that observed browsing patterns are significantly different from those previously assumed. Users' behavior frequently exhibits defensive strategies, such as multiple websites overlapping and downloading simultaneously, which can interfere with website fingerprinting. A study of international users demonstrated that over 15% of websites overlap with at least another, with an average overlap time of 66 seconds, while a US-based study showed only 0.72% of websites overlap. Moreover, these overlaps typically occur shortly after the initial website download. These findings suggest that the beginning of a website is more crucial than the end for website fingerprinting attacks, highlighting the need for more analysis of webbrowsing behavior.

Received on 08 May 2025; accepted on 20 January 2026; published on 22 January 2026

Keywords: Webbrowsing, Website Fingerprinting, Anonymity, Privacy.

doi:10.4108/eetss.9271

1. Introduction

Privacy is becoming an increasing concern for many. VPNs, Tor, and DuckDuckGo are increasingly being used. Many companies, such as Apple and Brave, are touting more private solutions. Users thus want to hide their browsing activities so that they cannot be tracked. This includes clearing browser cache, hiding their IP address and the server's IP address, and installing privacy tools such as ad-blockers. However, nothing is 100% secure or private. Even with all these measures in place, network traffic data can still be collected by a passive adversary. Even though network traffic is encrypted, the size of each network packet, the number of packets, and the direction of the packets (from client to server or from server to client) can still be seen. Furthermore, if the adversary is at the first hop of the victim, e.g. the Internet Service Provider (ISP), then that adversary knows the identity of the victim since the victim is directly connected to the ISP. The adversary still needs to determine the identity of the server/destination based only on the packet size and number of packets sent. This is an attack known as website fingerprinting.

Website fingerprinting has been known for a while [1], and much of the work in this area has improved the accuracy and feasibility of the attack. Website fingerprinting defenses have also been proposed and implemented in Tor [2]. However, most of the work have made some assumptions about the user. For example, only one website is visited at a time, and it can be determined when a website download starts and ends. Moreover, all the website fingerprinting data have been collected in laboratory settings.

The motivation for this work is that nobody has tried to determine whether website fingerprinting

*Corresponding author: Eric Chan-Tin. Email: dchantin@luc.edu

attacks are possible in a realistic setting based on real users' webbrowsing behaviors. The goal of this research is to find real users' webbrowsing behaviors in a privacy-preserving manner. This is done through A webbrowser extension called WebTracker was developed to track which websites users are visiting, when a website download starts, and when a website finishes downloading. Participants were then recruited and compensated to complete an anonymous survey and install WebTracker. The results could lead to more practical website fingerprinting attacks and countermeasures.

Two studies were conducted: 1) in 2020 with English-speaking international users, and 2) in 2022 with English-speaking users from the USA. The first study was to obtain a broad demographics and geographics diversity while the second study focused on the US only. The results from WebTracker show that almost 16% in 2020, and roughly 0.72% in 2022 of websites visited had an overlap with at least one other website. An overlap means that two websites visits are happening at the same time, thus that makes website fingerprinting harder as it cannot be determined whether a network packet is for website1 or for website2. This result shows that it is important to determine real webbrowsing behaviors and use that information to perform more realistic website fingerprinting attacks and to design more efficient defenses.

The contributions of this research are as follows.

- Empirically determine real users' browsing behaviors.
- Develop a privacy-preserving webbrowser extension, called WebTracker, to track user's webbrowsing activities.
- Evaluation shows that a non-trivial part of website downloads overlap with other websites, which could make website fingerprinting harder.
- Conducted two studies – once in 2020 for 211 days with 238 participants (installed 83 times), and once in 2022 for 426 days with 942 participants (installed 200 times).

This research is novel and noteworthy because there has been a lack of work in looking at how users browse the web in the context of website fingerprinting attacks. Previous work have looked at multi-page browsing [3], different geographical locations [4], different training/testing datasets [5], and the addition of noise [6, 7] This further adds to the literature of the realistic applications of website fingerprinting attacks because users' webbrowsing behaviors could act as a natural deterrent to website fingerprinting attacks. Understanding how real users browse the web and integrating that behavior into experiments can lead to more realistic findings on the practicality of website fingerprinting attacks. Although collecting real users' actual browsing information is problematic, doing an experimental lab study with no user behavior is also problematic. Thus, integrating user browsing behavior into future experimental studies can improve the understanding of website fingerprinting attacks in the wild. The first step is to obtain how users browse the web, which is the contribution of this research.

A previous version of this paper was published at [8]. This extended version contains an extra set of experiments performed in the USA and explains that website fingerprinting attacks are more realistic now due to faster Internet speeds (0.87 seconds median download time compared to 3.57 seconds median download time). The new set of experiments recruited almost 4 times more users (942 participants vs 238 participants) and lasted for much longer (14 months vs 7 months). This provides a much more comprehensive view of webbrowsing behaviors from real users.

2. Background and Related Work

To obtain user's webbrowsing behaviors, real users need to be recruited. Crowdsourcing platforms such as MTurk [9] and Microworkers [10] are used by researchers and companies to obtain real users to perform various tasks, such as completing a survey, testing some website features, or reviewing an application. Anybody can become a user of these crowdsourcing platforms. They may sign up for tasks and are compensated for their time. Users can also be recruited through social media platforms. In our case, we recruited participants from Microworkers to fill out an anonymous survey hosted by Qualtrics and then to install WebTracker on their computers. Microworkers was chosen over MTurk due to the ease of asking participants to download software. Since participants are anonymous, we could not identify who installed our extension and compensated everyone who completed the survey. Qualtrics includes a feature that does not log any IP addresses.

To record webbrowsing, a webbrowser extension was developed as an extension is lightweight and can access the internals of a webbrowser without being a full blown application. Any user of a computer can easily (un)install a webbrowser extension.

Website fingerprinting [1, 2, 11–17] is an attack that attempts to identify the website visited based only on network traffic metadata. That metadata consists of the network packet sizes, the direction of network packets, and the timing information of the packets. Recent experimental results have shown that website fingerprinting attacks can be accurate in predicting the website visited with over 95% accuracy. This is in either a closed world with 1,000 or more websites or

open world experiment with over 100,000 websites. Other fingerprinting attacks [18] exist but do not rely on network traffic and are orthogonal to this research. The passive adversary in a website fingerprinting attack sits on the first hop of the victim, e.g. the ISP or a Tor [19] entry relay. Knowing the victim, the goal of the adversary is to determine the website visited given that the IP address is hidden, e.g. using Tor or a VPN. Thus, only the size of each network packet, the number of packets, the direction of each packet (from server to client or from client to server), and the timing information of each packet are observed. Using only this information and machine learning algorithms such as k-nearest neighbor (K-NN), SVM, or deep learning, it has been shown that a website can be predicted with an accuracy over 94%.

The dataset used by the majority of previous work has been in a laboratory setting, where the authors collect the network traffic from a pre-determined set of websites using their lab machines. This is not a realistic setting, as regular users browse the web differently than going to website1, wait a certain amount of time, close the tab, and then go to website2, wait a certain amount of time, close the tab, and then go to website3, etc. More recent work [20] has looked at real browsing traffic on Tor — the authors showed that the accuracy decreases significantly when monitoring more than 25 websites. This work complements that paper by looking at how users browse the web to determine if website fingerprinting is still practical. Due to the complexity of asking users to install Tor, participants used their typical webbrowsers, such as Firefox or Chrome, with no Tor configuration. Another work [5] looks at a realistic attack on Tor using website fingerprinting, such as different training/testing datasets, multiple pages, and considering the effect of noise. This work differs as it looks at how users browse the web and creates a study based on the webbrowsing behavior. Another work [21] proposes a camouflage countermeasure by loading a page in the background. This work's goal is not looking at website fingerprinting defenses but whether users' webbrowsing behavior might help or hinder website fingerprinting attacks.

Much of the work [22–27] on website fingerprinting has collected data from one website at a time, which means a user would browse the web visiting one website at a time. Most of the work also clears the webbrowser cache between website visits. This research builds on [28] by further examining the impact of real webbrowser behaviors on website fingerprinting attacks. This research attempts to quantify how a user browses the web and whether that makes website fingerprinting attacks more or less feasible. Although some research [3, 29, 30] has relaxed some of these assumptions by looking at multiple tabs being opened and overlapping website visits, there has been little work on how users actually browse the web. This research looks at real users' webbrowsing behaviors, how many websites are visited, and how often they visit each website. This research will impact the practicality and feasibility of website fingerprinting attacks and defenses.

There have been previous work on monitoring user's webbrowsing behaviors [31–36]. Although previous work focused on creating a user browsing model or predicting which websites users will visit or determining the websites visited, this work focuses on whether users browse multiple websites at the same time. Tab Logger [37] is a webbrowser extension for users to look at their tab usage such as number of open tabs and the lifetime of each tab. Although similar, the WebTracker extension keeps track of when a website starts and finishes downloading as opposed to how long a tab is open. Studying how web users utilize the web [35] is the one of closest work to ours. Participants were recruited to have all their client traffic from Firefox captured. The purpose of that work is to look at how the web is used, and it was found that users utilize multiple tabs and do not utilize the back button that often. Our goal is to look at overlapping websites visits which was not looked at in [35]. Another work [36], that is close to ours, looks at tracing how users get to a malicious URL. They asked users to install a webbrowser extension (similar to us) and collected similar data as ours. However, their goal was different as that work was looking at tracing how a malicious URL was visited, whereas our goal is to look at whether users open multiple tabs at once and visit multiple websites at once. While designing the experimental data collection, [38] advice on conducting an ethical web study was followed. Table 1 shows a comparison of WebTracker with previous work.

3. Design

To track real users' webbrowsing behaviors, a privacy-preserving webbrowser extension called *WebTracker* was developed. WebTracker tracks when a user visits a new webpage and when the webpage finishes downloading. An overview of WebTracker is given in Figure 1. The tool tracks 1) when a user visits a new webpage, either by typing a URL and clicking *Enter* or by clicking on a URL link; 2) when a webpage finishes downloading; and 3) when a webbrowser tab is closed. Item #3 is not important to track the webbrowsing behaviors of users. The data collected are as follows.

- *userID*: a number that is randomly generated when WebTracker is installed. The number stays the same even if WebTracker is enabled/disabled. The number will be regenerated if the user uninstalls and reinstalls WebTracker. This is to track each user.

Table 1. Comparison of previous work with WebTracker.

Paper	Goal	Methodology	Dataset
[31]	Predict next webpage that a user will visit	Modified Markov Model	11,000 webpages over 8 months
[32]	Find web users' interest from browsing behavior	Time-based approach with known attributes	96 participants for four days
[33]	Network traffic generator	Models time of request and response for each object, not just website	Web logs of 62,000 clients for 3 days
[34]	Browsing behaviors but no timing information	Large-scale web panel data	Web histories of 250,000 individuals
[35]	Study of web use	Network proxy, which captured all client traffic from Firefox	25 participants over 3 months
[36]	Trace how users get to a malicious URL	Webbrowser extension	Approximately 1,500 users per month over 12 months
[37]	Tab usage	Webbrowser extension	No data collected
WebTracker	User browsing behavior especially when websites download starts and ends, and whether multiple tabs are open	Webbrowser extension	1,200 users over 20 months

- *tabID*: the tab number on which an action was triggered.
- *URL*: the second-level domain of the webpage.
- *status*: start downloading (0), finish downloading (1), or tab closed (2).
- *timestamp* (in UTC).

Every time an action is triggered, the information above is concatenated as a comma-separated string and sent as a POST request to our webserver. We emphasize that the tool does not store any data locally. Every action is sent as an encrypted POST packet to our webserver. An eavesdropper could potentially determine that a user has installed WebTracker but this does not reveal the website visited.

To measure the start/finish of a page download, the *chrome.tabs* API for Chrome and the *browser.tabs* API for Firefox are used. These APIs have a method *onUpdated*, which is called whenever a tab is updated in any way. The method has a type, *changeInfo*, with subtype "status" that returns the status of the tab. For our purposes, the relevant return values were "loading" and "complete" which are codes of "1" and "2", respectively. The timestamp is also provided —this way we were able to determine the amount of time taken for the tab to go from "loading" to "complete". Whenever a tab is closed, the *chrome/browser.tabs* API fires an event *onRemove* — the event has a type *TabID*, which is used to track which tab was closed. We would check to ensure that the tab being closed was actually a website, and not just a blank tab.

To protect the privacy of users, the webserver is set to not record any IP address. Thus, the only information recorded is the information sent from the webbrowser. To further protect the privacy of users, each URL is hashed. The webserver is only publicly accessible through port 443.

WebTracker was implemented as an extension for two webbrowsers: Google Chrome and Mozilla Firefox. Since they are extensions, they utilize HTML (~40 lines of code), CSS (~100 lines of code), and JavaScript (~200 lines of code). The implementation is straightforward and contains a simple ON/OFF switch for users to choose when they want their webbrowsing behavior tracked or not. jQuery, a JavaScript library, is used for the switch to work. To avoid accidental downloads of the WebTracker, we did not post the extension on the official Chrome web store or Firefox add-on extension site. To install the webbrowser extension in either Firefox or Chrome, a user needs to manually download the extension from our webserver and manually install it.

The functionality begins with the application recognizing if the extension is on or off, as shown in Figure 1. If it is off, it does nothing. If it is on, the extension utilizes JavaScript listener events to listen for when Chrome or Firefox fires a trigger that something changed on the browser. Then the data (i.e., *userID*, *tabID*, *URL*, *status*, *timestamp*) is collected and then sent

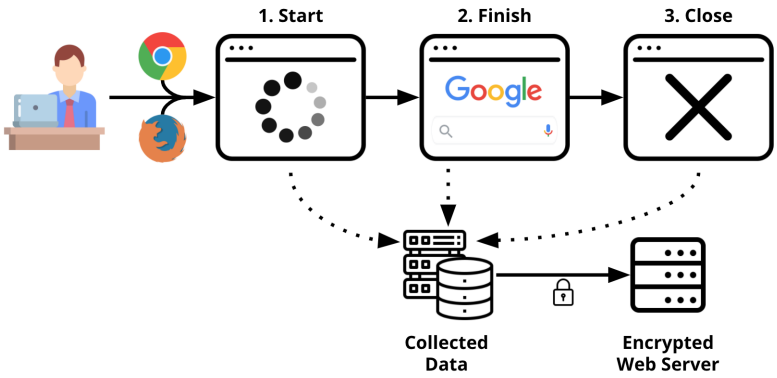


Figure 1. Overview of WebTracker.

encrypted to our webserver. The webserver creates a log of all that information that is sent.

4. Evaluation

4.1. Data Collection: 2020

WebTracker was active, collecting data, from June 1, 2020 to December 31, 2020. Since we did not know who installed WebTracker, we could not ask the participants to uninstall the extension. Thus, on January 1, 2021, WebTracker automatically disabled itself. Users were recruited from Microworkers [10] crowdsourcing platform from June 1, 2020 to September 1, 2020. This allowed for four months of data recorded for the participants recruited on September 1 and seven months of data recorded for the participants recruited on June 1. Participants could disable or uninstall WebTracker at any time. IRB (Institutional Review Board) approval was obtained before the experiments were conducted.

Participants recruited from Microworkers had to be over 18 years old, had to understand English and can be from any country. We could not verify this information since we do not know who the workers were, but these were options selected on the Microworkers site. Every participant was asked to fill out a brief survey. The survey described the experiment and asked for their worker ID. Before filling out the worker ID, participants were asked to install WebTracker. They were provided installation and uninstallation instructions for both Mozilla Firefox and Google Chrome. Each participant was compensated with $1.50 after completion of the survey. Due to the privacy features implemented in WebTracker, it cannot be determined whether a participant installed the extension. A total of 238 participants were recruited.

4.2. Data Collection: 2022

For our second study, WebTracker was active, collecting data, from May 1st, 2021, to July 1st, 2022. All Microworkers participants were based out of the United States. This was to maintain consistency and minimize the skewing of our data due to users with poor Internet connections. A total of 942 participants were recruited.

4.3. WebTracker Results

Browsing Results.

Recall that the goal of WebTracker is to privately collect participants' web browsing behavior in order to garner insight on real-world website visit overlaps, which can be used further in website fingerprinting research. The data collected includes a unique and random userID so that each individual user can be monitored, the tabID in case a user visits the same website using different tabs, the URL, the status so that we know when website download starts and stops, and the timestamp of that event. Users can enable/disable WebTracker at anytime. Moreover, webbrowsers could crash or be closed before all websites have finished downloading. Finally, WebTracker could have been installed while other website visits are in progress. It is highly unlikely that a user will install WebTracker on a fresh webbrowser. Thus, our data collection was conservative: 1) we excluded any URLs that did not have a start downloading or finish downloading status; 2) we excluded any website visits that took longer than 10 minutes to download. It is atypical for a webpage to take more than several seconds to finish downloading. That large download time could be because it was a visit to the same URL on the same tab but at a later time. As an example, a participant could start to visit a website on the first tab and then disable WebTracker. The website finishes downloading. Some time later, the same participants visits the same website on the first

(same) tab, then enables WebTracker. The extension will log when that website finishes downloading. The time difference for that URL will be big since WebTracker only knew about the start of the first visit and the end of the second visit; 3) we excluded data that came from the *microworkers.com* website. The reason is that our participants were recruited from that website and likely visited it often, potentially skewing the data.

Study 1: 2020

WebTracker was installed a total of 83 times. This is lower than the 238 users recruited because not all users installed the extension. Due to the way the experiment was setup, it was not possible to determine which user installed the extension. Moreover, some users uninstalled the extension soon after install because we only obtained data from some userIDs for only a few minutes. The total time elapsed between the start of our data collection and the end is about 211 days (almost 7 months from June 1 to December 31). The total number of websites counted in this entire dataset is 57,097 websites, or roughly about 270 websites visited a day on average. Out of that time, users spent 50.88% of that time downloading data, or about 107 days. Furthermore, out of all the time spent downloading, 15.66% of downloaded websites were overlapping, with 2.11 websites in an overlap on average.

Figure 2 shows that up to 75% of websites take less than 10 seconds to download. Some websites do take longer to download. This could be possible because websites that users visit could include downloading files such as photos, videos, and music, which are often large in file size and take longer to download. Moreover users that participated in this research might have come from different countries, meaning some may not have access to high internet speeds, which may result in a longer download time. The average download time was 16.68 seconds, and the median download time was 3.57 seconds with a standard deviation of 47.95 seconds. This indicates that the data itself varies greatly, but the low median suggests most websites do not take long to download, with larger websites bringing up the average. A larger variance in website downloads potentially indicates that websites are more unique. If a website took a longer time to download than usual, its packet data could point to it belonging to a media website, such as one with videos or pictures. Figure 3 shows the download time only for websites that are overlapping with other websites. As shown in the figure, the median download time is 12.5 seconds, which is much higher than the median download time of 3.57 seconds for all websites. This makes sense as websites that take longer to download (for example, downloading a movie) are more likely to overlap with other websites as the user opens a different tab. Furthermore, Figure 4 includes the websites visits of the 83 users that participated in this research. Some users only visited a few sites while using WebTracker, but the greater proportion visited tens or hundreds of websites, with 50% of users visiting at least roughly 50 websites.

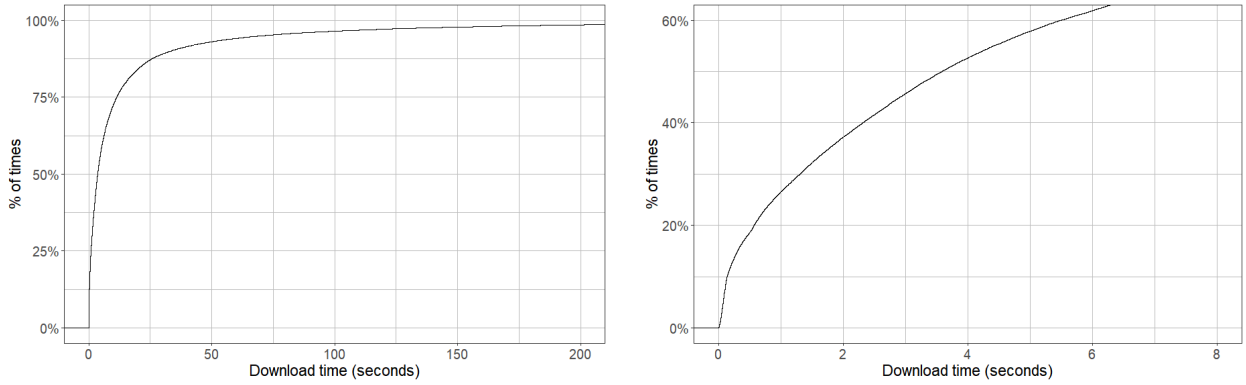
Study 2: 2022

WebTracker was installed a total of 200 times. This is lower than the 942 users recruited for the same reason as the first study in 2020. The total time elapsed between the start of our data collection and the end is about 426 days (almost 14 months from May 1st, 2021 to July 1st, 2022). The total number of websites successfully downloaded in this dataset is 108,152 websites, or roughly about 254 websites visited a day on average. Out of that time, users spent 0.66% of that time downloading data, or about 2.82 days. Downloading time reflects how long it took to download a page and does not reflect the time spent browsing. Furthermore, out of all the time spent downloading, 0.72% of downloaded websites were overlapping, with 2.19 websites in an overlap on average.

Figure 5 shows that up to 75% of websites take less than 2.36 seconds to download. Some websites do take longer to download. This could be possible because websites that users visit could include downloading files such as photos, videos, and music, which are often large in file size and take longer to download. The average download time was 2.26 seconds, and the median download time was .87 seconds with a standard deviation of 7.06 seconds. This indicates that the data itself varies greatly, but the low median suggests most websites do not take long to download, with larger webpages or files bringing up the average. A larger variance in website downloads potentially indicates that websites are more unique. If a website took a longer time to download than usual, its packet data could point to it belonging to a media website, such as one with videos or pictures. Figure 6 shows the download time only for websites that are overlapping with other websites. As shown in the figure, the median download time for overlapping websites is 3 seconds, which is higher than the median download time of 0.87 seconds for all websites. Furthermore, Figure 7 includes the websites visits of the 200 users that participated in this research. Some users only visited a few sites while using WebTracker, but the greater proportion visited tens or hundreds of websites, with 50% of users visiting at least roughly 10 websites.

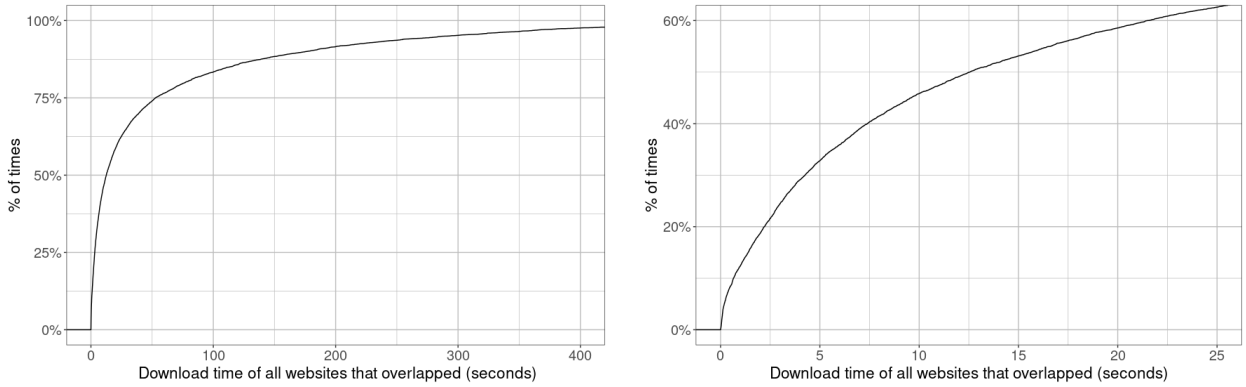
Overlapping Results.

An overlap occurs when two or more websites are being downloaded at the same time. Determining when an overlap ends is more involved. Figure 8 shows a diagram of a hypothetical webbrowsing history for a user. The user visits three websites: Website1 starts at t1 and ends at t6, Website2 starts at t2 and ends at t4, and Website3 starts at t3 and ends at t5. The overlap time in the figure is t5-t2 and encompasses all three websites.



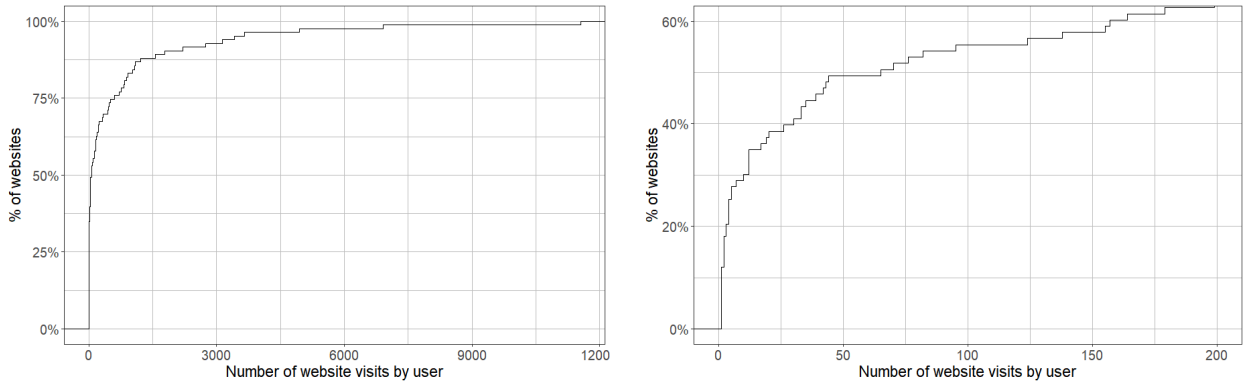
(a) (b)

Figure 2. Empirical Cumulative Distribution Function (CDF) plot of time taken to download a website. (a) shows the CDF for download times up to 200 seconds and (b) shows a zoomed-in figure with the CDF percentage up to 60%.



(a) (b)

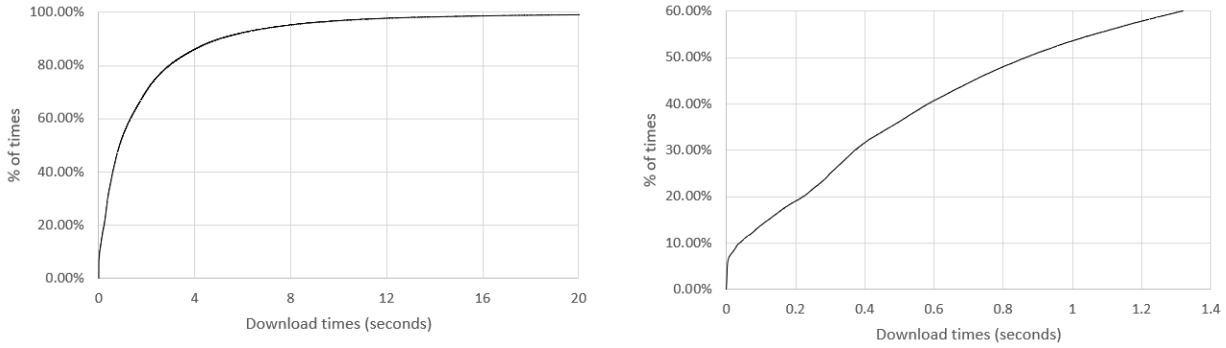
Figure 3. Empirical cumulative distribution function (CDF) plot of download times of only websites that have overlapped. (a) shows the CDF for loading times up to 200 seconds and (b) shows a zoomed-in figure with the percentage up to 60%.



(a) (b)

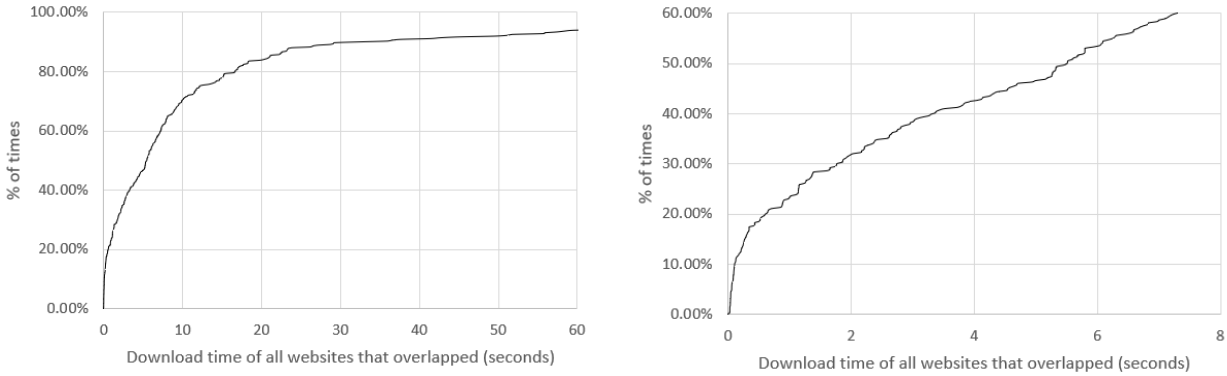
Figure 4. Empirical cumulative distribution function (CDF) plot of number of websites visited per user. (a) shows the CDF for up to 1,200 website visits (b) shows a zoomed-in figure with the CDF percentage up to 60%.

Though more than two websites are overlapping, the entire overlap time is taken as the elapsed time when all involved websites are overlapping each other. Therefore, Website2 and Website3 are not counted as their own overlap, given that they overlap already with Website1 – the three websites are counted as one



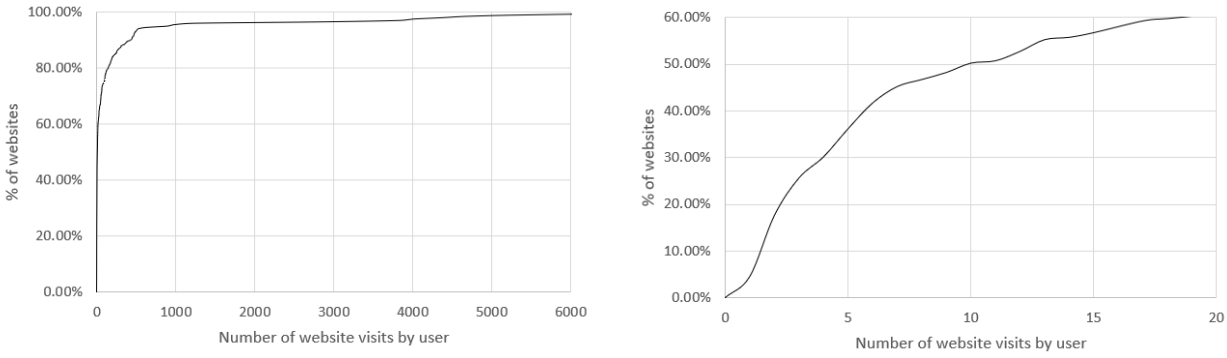
(a) (b)

Figure 5. Empirical Cumulative Distribution Function (CDF) plot of time taken to download a website. (a) shows the CDF for download times up to 20 seconds and (b) shows a zoomed-in figure with the CDF percentage up to 60%.



(a) (b)

Figure 6. Empirical cumulative distribution function (CDF) plot of download times of only websites that have overlapped. (a) shows a zoomed-in figure with the download times up to 60 seconds and (b) shows a zoomed-in figure with the CDF percentage up to 60%.



(a) (b)

Figure 7. Empirical cumulative distribution function (CDF) plot of number of websites visited (successfully downloaded) per user. (a) shows the CDF for up to 6,000 website visits (b) shows a zoomed-in figure with the CDF percentage up to 60%.

overlap. However, not all websites were downloading across the entire overlap, such as Website3 was not present for time t2 to t3. We instead averaged the number of active websites in the overlap: first, there were 2 websites in the overlap between t2 and t3, then 3 websites between t3 and t4, then 2 websites between t4 and t5. Therefore, the number of websites in this overlap is $(2 + 3 + 2)/3 = 2.33$. Another important metric is the elapsed time before overlap occurs, which in the diagram is t2-t1. This shows the time when

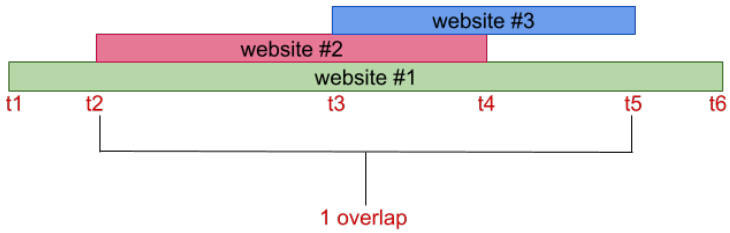


Figure 8. Diagram demonstrating how an overlap is counted. It is not contingent on the number of websites present, but rather on when all websites that are overlapping have finished. Therefore, though there are more than two websites, the entire bundle of websites is counted as one overlap. The overlap time is thus t5 – t2. The elapsed time before overlap is t2 – t1.

Website1 is downloading before other websites start. This is important as a longer t2 - t1 time means a possibly easier prediction of Website1 because more packets of Website1 are unaffected by other websites. The last metric is download time. The download time for Website1 is t6 - t1. The download time for Website2 is t4 - t2. The download time for Website3 is t5 - t3.

Figure 9 shows how our algorithm works in calculating overlaps. The active list keeps track of all websites that are still downloading (status "1"). Websites are removed from the active list when they are finished downloading; those will have a status of "2". If a website starts downloading and the active list is empty, then the start of a new overlap is found. If a new website starts downloading and the number of websites in the active list is greater than 1, then an overlap is occurring and is measured accordingly. When a website finishes downloading and the number of websites in the active list decreases back to 1, then overlapping has stopped.

Study 1: 2020

Figure 10 shows the CDF for the overlap times, with the average length of an overlap being 223.19 seconds and the median time being 66.68 seconds with a standard deviation of 836.07 seconds. This indicates that the data for the overlap time varies significantly. This suggests that the overlap length is dependent on user activity, whether they are downloading a large file while browsing other websites, watching multiple videos, streaming, or performing multiple small tasks at the same time. Long overlaps could potentially produce enough noise in the data to make it difficult to fingerprint the website. If the overlap is long, it would suggest that the websites are overlapping during the duration of their entire downloading period, rather than a small overlap where websites overlap momentarily. Figure 12 show the elapsed download time of the first website before overlap occurs. 25% of the time, the overlap occurs after 4 seconds of download. The average amount of time a website downloads before overlapping occurs is 24.06 seconds with a median of 8.738 seconds and standard deviation of 45.62 seconds. Additionally, Figure 11 demonstrates the total elapsed download time of the first website in an overlap. The average amount of time it takes for the first website to download is 97.92 seconds, with median 41.46 seconds with a standard deviation of 127.24. The long elapsed times from Figure 12 indicate it generally takes some time before the first website overlaps with another website. That can pose an issue as most website fingerprinting algorithms look at the whole website network trace rather than just the first few packets or seconds. Therefore, an earlier overlap means it might be more difficult to perform a website fingerprinting attack. As can be seen from Figure 11 and Figure 12, it takes almost 9 seconds before overlap occurs. Moreover, the first website, when overlap occurs, takes 41 seconds to download. This means that 22% of the first website is not overlapping with any other website.

Study 2: 2022

Figure 13 shows the CDF for the overlap times, with the average time of an overlap being 4.76 seconds and the median time being 1.49 seconds with a standard deviation of 13.02 seconds. This indicates that the data for the overlap time varies significantly. This suggests that the overlap time is dependent on user activity, whether they are downloading a large file while browsing other websites, watching multiple videos, streaming, or performing multiple small tasks at the same time. Long overlaps could potentially produce enough noise in the data to make it difficult to fingerprint the website. If the overlap is long, it would suggest that the websites are overlapping during the duration of their entire downloading period, rather than a small overlap where websites overlap momentarily. Figure 15 show the elapsed download time of the first website before overlap occurs. 60% of the time, the overlap occurs after 3.43 seconds of download. The average amount of time a website downloads before overlapping occurs is 11.82 seconds with a median of 2.23 seconds and standard deviation of 48.69 seconds. Additionally, Figure 14 demonstrates the total elapsed download time of the first website in an overlap. The average amount of time it takes for

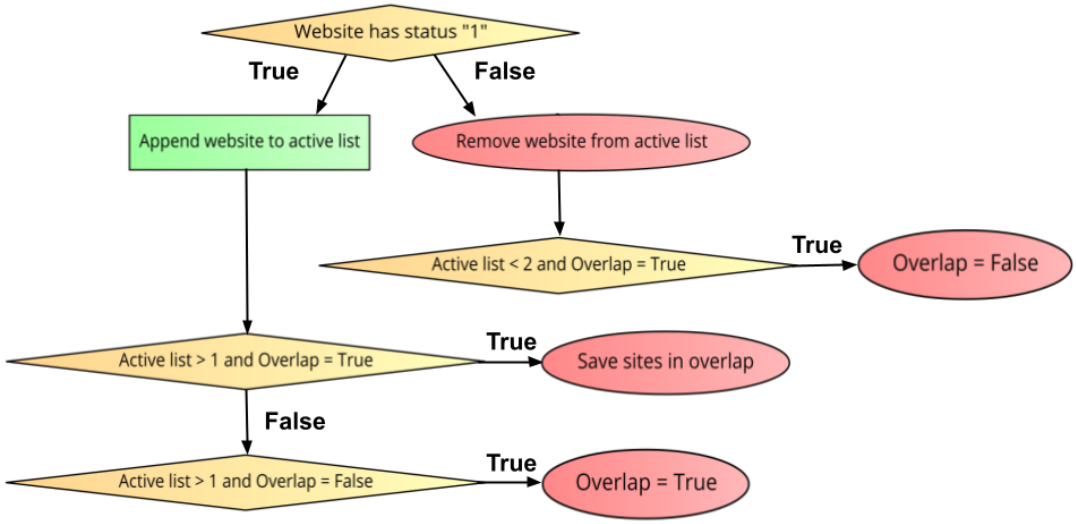
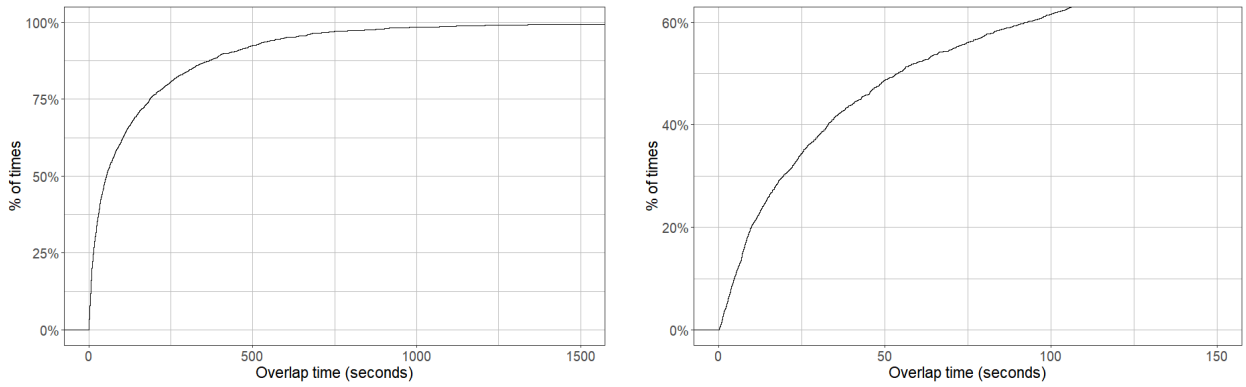
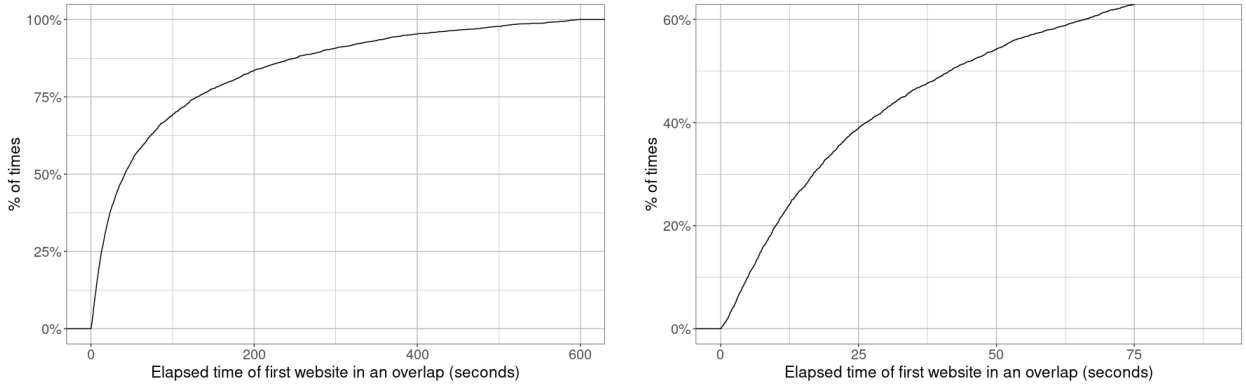


Figure 9. Flowchart showing how our algorithm works in calculating overlaps.



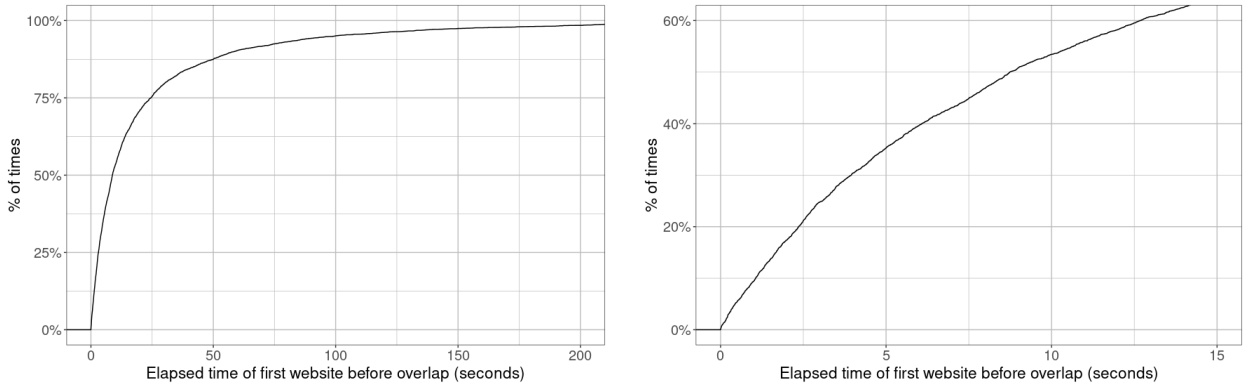
(a) (b)

Figure 10. Empirical cumulative distribution function (CDF) plot for overlap times. (a) shows the CDF for overlap times up to 1,500 seconds and (b) shows a zoomed-in figure with the CDF percentage up to 60%.



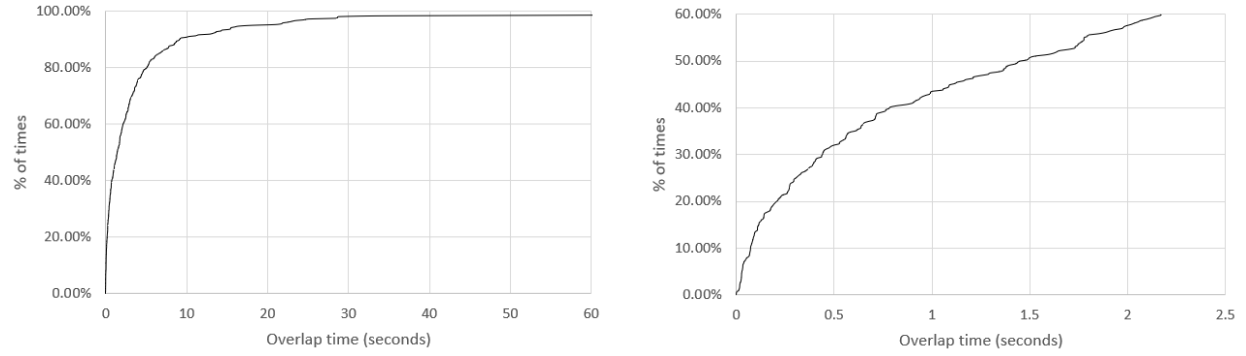
(a) (b)

Figure 11. Empirical cumulative distribution function (CDF) plot of time it takes for the first website in an overlap to download. (a) shows the CDF for loading times up to 200 seconds and (b) shows a zoomed-in figure with the percentage up to 60%.



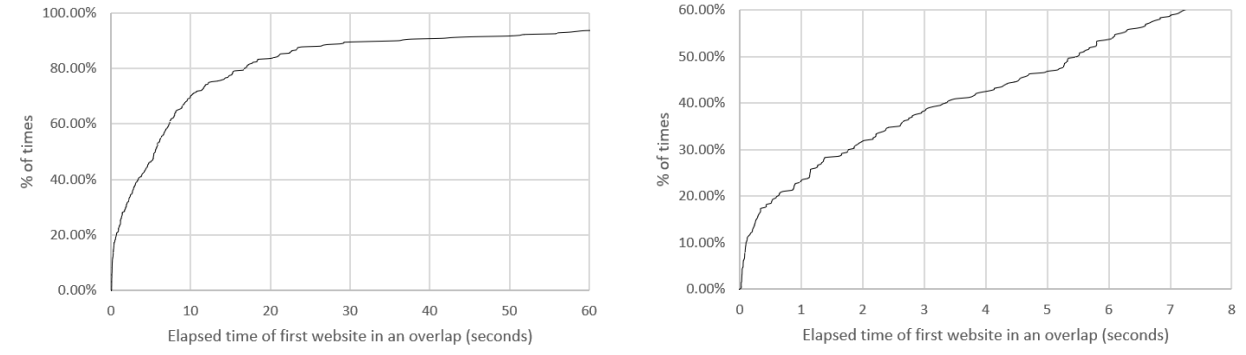
(a) (b)

Figure 12. Empirical cumulative distribution function (CDF) plot of time it takes for the first website in an overlap to download before the overlap occurs. (a) shows the CDF for loading times up to 600 seconds and (b) shows a zoomed-in figure with the percentage up to 60%.



(a) (b)

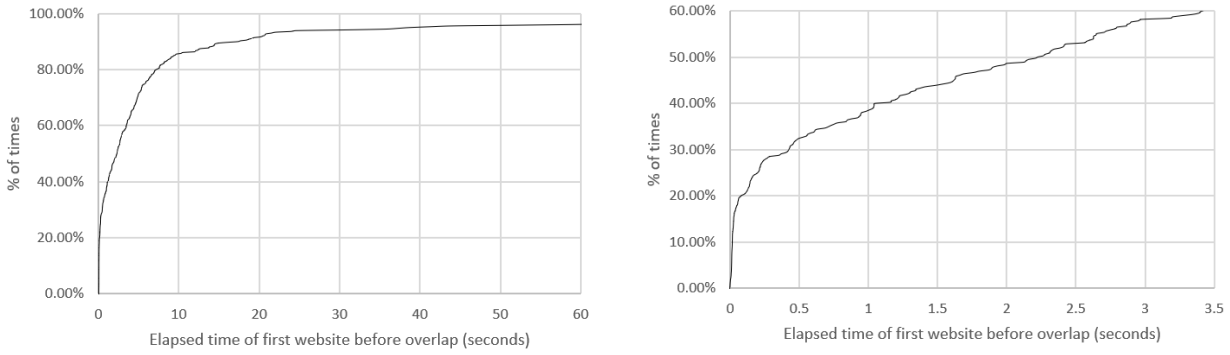
Figure 13. Empirical cumulative distribution function (CDF) plot for overlap times. (a) shows the CDF for overlap times up to 60 seconds and (b) shows a zoomed-in figure with the CDF percentage up to 60%.



(a) (b)

Figure 14. Empirical cumulative distribution function (CDF) plot of time it takes for the first website in an overlap to download. (a) shows the CDF for loading times up to 60 seconds and (b) shows a zoomed-in figure with times up to 60%.

the first website to download is 19.46 seconds, with a median of 5.51 seconds and a standard deviation of 61.45. The long elapsed times from Figure 15 indicate it generally takes some time before the first website overlaps with another website. That can pose an issue as most website fingerprinting algorithms look at the whole website network trace rather than just the first few packets or seconds. Therefore, an earlier overlap



(a) (b)

Figure 15. Empirical cumulative distribution function (CDF) plot of time it takes for the first website in an overlap to download before the overlap occurs. (a) shows the CDF for loading times up to 60 seconds and (b) shows a zoomed-in figure with times up to 60%.

means it might be more difficult to perform a website fingerprinting attack.

Summary

The results for our two studies are summarized in Table 2. Recall that Study 1 was by English-speaking users from any country while Study 2 was from users located in the United States. The results vary greatly, likely because users from the United States have a faster Internet connection – this is depicted by the much smaller median download time for all websites (0.87 seconds). The number of overlaps in Study 2 was also small and when an overlap does occur, it was short (1.49 seconds in Study 2 compared to 66.68 seconds in Study 1). The results indicate that overlapping could have an effect on website fingerprinting attacks, especially if the models considers the whole website instead of the first few network packets. A slower Internet speed tend to increase the number of overlaps, which in turn could increase the protection of the users in a website fingerprinting attack.

4.4. WebTracker Experiment

To evaluate the webbrowsing behaviors, we set up an experiment where we collected the network traffic from the top 100 websites according to Alexa top sites [39]. We collected 20 samples for each of the top 100 websites using tcpdump to collect the network traffic. Previous work [40, 41] have used 20 or 40 samples and Random Forest, SVM, or K-NN – we found Random Forest with 20 samples was enough for our study. We used the Mozilla Firefox webbrowser since the users we surveyed utilize a regular webbrowser (not Tor) and we had no proof that they were running a VPN. The goal of this experiment is to simulate the webbrowser behavior of a user (visiting more than one website at a time) and its impact on website fingerprinting. To simulate the webbrowsing behavior, we ran two more experiments: 1) a new tab is opened visiting a random website after an average of 2.23 seconds, to mimic the results found in the US dataset, and 2) a new tab is opened visiting a random website after an average of 8.74 seconds, to mimic the results found in the international dataset. We used the Random Forest machine learning classifier with the first 2,500 packet sizes as features. This is similar to previous work [17, 40]. We found that the accuracy of identifying the correct website decreases by 26.8% when the overlap occurs after 2.23 seconds and the accuracy decreases by 25.67% when the overlap occurs after 8.74 seconds. This means that having an overlap does cause a decrease in accuracy – however, there is not much difference between the two overlap experiments. This is likely because the data was collected in a network with fast Internet connectivity, so it did not matter whether the second website occurs after 2 seconds or 8 seconds. We found the base accuracy with no overlap to be 38.3%; although this is lower than previous work [41] of 90%, this is still 38 times higher than a random guess of 1/100 = 1%. The goal of this experiment is to compare the accuracy between no overlap and overlapping websites, not to find the best model for website fingerprinting attacks. Our conclusion is not affected.

5. Discussion

The results show that users' browsing habits are different than assumed. We found in 2020 that occasionally even four websites can download at the same time. In 2022, we found that at most two unique websites overlapped downloading at the same time. The rest of the overlaps consisted of the same website attempting to download in another tab. This adds a layer of complexity to determining what website(s) a network trace belongs to. Additionally, in 2020, overlaps, though varying greatly in length, tended to be long, with a median of 66 seconds. In 2022, overlaps tended to be short, with a median of 1.49 seconds.

Table 2. Summarized Results for Study1 and Study2.

	2020	2022
Number of participants	238	942
Number of webtracker installs	83	200
Number of days of data collected	211	426
% of websites that overlap with at least one other website	15.66%	0.72%
Median download time for websites (seconds)	3.57	0.87
Median download time for overlapping websites (seconds)	12.5	3
Median time of an overlap (seconds)	66.68	1.49
Median time before an overlap starts (seconds)	8.738	2.23

Long overlaps could be a potential countermeasure against website fingerprinting attacks as overlapping websites are more difficult to identify. However, users only spent 15.66% of their downloading time in overlap in 2020, and 0.72% of their time in 2022, meaning that a majority of users' website visits could potentially be identified. Additionally, we found that websites download a few seconds of data before overlap occurs, with a median of 8.738 seconds elapsed time in 2020, and 2.23 seconds elapsed time in 2022. Previous research have devised an algorithm that can accurately predict a website with only the first few seconds of its packet data with about 85% accuracy [42]. This is because websites are easier to identify from the beginning of the trace rather than the end of the trace. Therefore, though overlaps tend to be long, some websites may still be able to be identified if they are the first website in an overlap but the second website will be harder to identify. Based on this result, website fingerprinting models that utilize the whole network traffic are likely not reliable.

All the webbrowsing data collected are through a Firefox or Chrome extension. We did not measure whether the traffic is through a VPN or through the Tor network or whether the participants were using the Tor browser. Most web users do not use Tor, thus we doubt that any of the data collected is through the Tor network. This could be future research to measure only webbrowsing behavior on the Tor browser.

We claim our WebTracker to be "privacy-preserving"; however, it is still a tracker. We attempted to remove any identifiable information such as IP address and URL. We contacted our university's Institutional Review Board (IRB) and they deemed no IRB protocol was needed. We still consulted with the IRB to ensure that we are not collecting any privately identifiable information. Due to privacy issues, we did not collect the network traffic. This could have provided more insight on who is browsing how. Due to the lack of URLs and the lack of actual packet metadata, we could not do a website fingerprinting attack on the data we collected. It would have been interesting to determine the website fingerprinting accuracy on the overlapped websites.

We do not expect that the data collection sent to the server interferes with the data collection. The data sent is small (around 5KB including the TLS handshake) and is not recorded by the WebTracker extension.

It has been shown [43] that participants recruited online for a crowdsourcing platform are as reliable as in-person recruitment or a participants' panel. It is expected that this holds for webbrowsing behaviors because the participants use the web regularly and knew how to install a webbrowser extension. This is likely typical for a regular web user since they are browsing on a PC computer. Moreover, they know how to follow simple instructions to install an extension which most web users likely know, e.g. many user install ad-blockers, shopping tools like Rakuten, and so on.

There are some limitations. The first one is that participants were aware that their browsing activities were being monitored, thus the webbrowsing behavior might not be natural. Participants could also turn off/on the webbrowser extension at any time. Another limitation is that WebTracker did not have a mobile version and only recorded webbrowsing habits of desktop users of Firefox and Chrome. Therefore, users' mobile browsing habits are not known, nor is it known if browsing through apps rather than only through a dedicated browser affects overlaps. Moreover, there were only 283 installs of WebTracker. Due to the longer than normal download times of a few seconds, it could be that many of the participants had slower Internet connection or were not from the US or using a VPN. More users from more diverse locations need to be recruited. Future work might also explore requiring participants to upload a screenshot showing the extension has been installed before compensation.

6. Conclusion and Future Work

This research showed two main results: 1) privately track real users' webbrowsing behaviors such as how many websites they visit, how long each website takes

to download, and how many websites are visited at the same time; 2) show that website fingerprinting attacks make a strong assumption that users browse one website at a time and the adversary have access to all the network packets for each website visit. Actually, over 15% in 2020, and over 0.72% in 2022 of all website visits overlapped with at least one other website. Moreover, each overlap in 2020 lasted about 66 seconds, and 1.49 seconds in 2022.

With the way the participants in this research browsed the web, there were overlaps which in some way act as a countermeasure against website fingerprinting. Overlaps tend to decrease the accuracy of website fingerprinting attacks. Unfortunately, we could not record the number of packets and thus, could not determine the number of packets during the first non-overlapping few seconds of website download.

As future work, more participants and from more diverse locations need to be recruited and their webbrowsing behaviors analyzed. The network environment of each user also needs to be measured, such as bandwidth, latency, and geolocation. This work could also be extended to mobile devices such as smartphones and tablets. Recording webbrowsing behaviors on the Tor browser could also be future work. No website fingerprinting attack is performed – the next step would be to collect data by simulating the webbrowsing behavior observed and determine the effectiveness of website fingerprinting attacks.

7. Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. DGE 1918591 and 1919004. Project sponsored by the National Security Agency under Grant Number H98230-21-1-0325. The United States Government is authorized to reproduce and distribute reprints notwithstanding any copyright notation herein. This manuscript is submitted for publication with the understanding that the United States Government is authorized to reproduce and distribute reprints. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Security Agency.

References

[1] Hintz, A. (2003) Fingerprinting websites using traffic analysis. In *Proceedings of 2Nd International Conference on Privacy Enhancing Technologies*, PET'02 (Berlin, Heidelberg: Springer-Verlag): 171–178.

[2] Juarez, M., Imani, M., Perry, M., Diaz, C. and Wright, M. (2016) Toward an efficient website fingerprinting defense. In *ESORICS*.

[3] Jin, Z., Lu, T., Luo, S. and Shang, J. (2023) Transformer-based model for multi-tab website fingerprinting attack. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, CCS '23 (New York, NY, USA: Association for Computing Machinery): 1050–1064. doi:10.1145/3576915.3623107, URL https://doi.org/10.1145/3576915.3623107.

[4] Beckerle, M., Magnusson, J. and Pulls, T. (2022) Splitting hairs and network traces: Improved attacks against traffic splitting as a website fingerprinting defense. In *Proceedings of the 21st Workshop on Privacy in the Electronic Society*, WPES'22 (New York, NY, USA: Association for Computing Machinery): 15–27. doi:10.1145/3559613.3563199, URL https://doi.org/10.1145/3559613.3563199.

[5] Wang, T. and Goldberg, I. (2016) On realistically attacking tor with website fingerprinting. In *PETS*.

[6] Cui, W., Yu, J., Gong, Y. and Chan-Tin, E. (2018) Realistic cover traffic to mitigate website fingerprinting attacks. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*: 1579–1584. doi:10.1109/ICDCS.2018.00175.

[7] Cai, X., Zhang, X.C., Joshi, B. and Johnson, R. (2012) Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12 (New York, NY, USA: ACM): 605–616. doi:10.1145/2382196.2382260.

[8] Reyes, D., Dynowski, E., Chovan, T., Mikos, J., Chan-Tin, E., Abuhamad, M. and Kennison, S. (2023) Webtracker: Real webbrowsing behaviors. In *2023 Silicon Valley Cybersecurity Conference (SVCC)*: 1–8. doi:10.1109/SVCC56964.2023.10164930.

[9] Amazon Mechnical Turk (Accessed 2023), https://www.mturk.com/.

[10] Microworkers (Accessed 2023), https://www.microworkers.com/.

[11] Hayes, J. and Danezis, G. (2016) k-fingerprinting: A robust scalable website fingerprinting technique. In *25th USENIX Security Symposium (USENIX Security 16)*: 1187–1203.

[12] Cherubin, G., Hayes, J. and Juárez, M. (2017) Website fingerprinting defenses at the application layer. *PoPETs* **2017**(2): 186–203. doi:10.1515/popets-2017-0023.

[13] Oh, S.E., Li, S. and Hopper, N. (2017) Fingerprinting keywords in search queries over tor. *PoPETs* **2017**.

[14] Cherubin, G. (2017) Bayes, not naive: Security bounds on website fingerprinting defenses. *Proceedings on Privacy Enhancing Technologies* : 135–151.

[15] Kohls, K., Rupprecht, D., Holz, T. and Pöpper, C. (2019) Lost traffic encryption: Fingerprinting lte/4g traffic on layer two. In *Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*: 249–260. doi:10.1145/3317549.3323416.

[16] Rimmer, V., Preuveneers, D., Juárez, M., van Goethem, T. and Joosen, W. (2018) Automated feature extraction for website fingerprinting through deep learning. *25th Symposium on Network and Distributed System Security (NDSS)* .

[17] Sirinam, P., Imani, M., Juarez, M. and Wright, M. (2018) Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18: 1928–1943. doi:10.1145/3243734.3243768.

[18] Karami, S., Ilia, P. and Polakis, J. (2021) Awakening the web's sleeper agents: Misusing service workers for privacy leakage. In *Network and Distributed System Security Symposium*.

[19] Tor (2022), https://www.torproject.org/.

[20] Cherubin, G., Jansen, R. and Troncoso, C. (2022) Online website fingerprinting: Evaluating website fingerprinting attacks on tor in the real world. In *31st USENIX Security Symposium (USENIX Security 22)* (Boston, MA: USENIX Association): 753–770.

[21] Panchenko, A., Niessen, L., Zinnen, A. and Engel, T. (2011) Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the Workshop on Privacy in the Electronic Society (WPES 2011)* (ACM).

[22] Rahman, M.S., Sirinam, P., Mathews, N., Gangadhara, K.G. and Wright, M. (2020) Tik-tok: The utility of packet timing in website fingerprinting attacks. *Proceedings on Privacy Enhancing Technologies* **2020**(3): 5–24. doi:doi:10.2478/popets-2020-0043.

[23] Pulls, T. and Dahlberg, R. (2020) Website fingerprinting with website oracles. *Proceedings on Privacy Enhancing Technologies* **2020**(1): 235–255.

[24] De la Cadena, W., Mitseva, A., Hiller, J., Pennekamp, J., Reuter, S., Filter, J., Engel, T. *et al.* (2020) Trafficsliver: Fighting website fingerprinting attacks with traffic splitting. In *Proceedings of the ACM Conference on Computer and Communications Security*.

[25] Wang, T. (2020) High precision open-world website fingerprinting. In *IEEE Symposium on Security and Privacy (SP)*: 231–246. doi:10.1109/SP.2020.00015.

[26] Sirinam, P., Mathews, N., Rahman, M.S. and Wright, M. (2019) Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In *ACM SIGSAC Conference on Computer and Communications Security*, CCS '19: 1131–1148. doi:10.1145/3319535.3354217.

[27] Wang, C., Dani, J., Li, X., Jia, X. and Wang, B. (2021) Adaptive fingerprinting: Website fingerprinting over few encrypted traffic. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*: 149–160.

[28] Juarez, M., Afroz, S., Acar, G., Diaz, C. and Greenstadt, R. (2014) A critical evaluation of website fingerprinting attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14 (New York, NY, USA: ACM): 263–274. doi:10.1145/2660267.2660368.

[29] Xu, Y., Wang, T., Li, Q., Gong, Q., Chen, Y. and Jiang, Y. (2018) A multi-tab website fingerprinting attack. In *Proceedings of the 34th Annual Computer Security Applications Conference*, ACSAC '18 (New York, NY, USA: ACM): 327–341. doi:10.1145/3274694.3274697.

[30] Cui, W., Chen, T., Fields, C., Chen, J., Sierra, A. and Chan-Tin, E. (2019) Revisiting assumptions for website fingerprinting attacks. In *ACM Asia Conference on Computer and Communications Security*, AsiaCCS '19 (ACM). doi:10.1145/3321705.3329802.

[31] Awad, M.A. and Khalil, I. (2012) Prediction of user's web-browsing behavior: Application of markov model. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **42**(4): 1131–1142.

[32] Liang, T.P. and Lai, H.J. (2002) Discovering user interests from web browsing behavior: An application to internet news services. In *Proceedings of the 35th annual Hawaii international conference on system sciences* (IEEE): 2718–2727.

[33] Lee, J.J. and Gupta, M. (2007) A new traffic model for current user web browsing behavior. *Intel corporation* .

[34] Goel, S., Hofman, J. and Sirer, M. (2012) Who does what on the web: A large-scale study of browsing behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*, **6**.

[35] Weinreich, H., Obendorf, H., Herder, E. and Mayer, M. (2008) Not quite the average: An empirical study of web use. *ACM Transactions on the Web (TWEB)* **2**(1): 1–31.

[36] Takahashi, T., Kruegel, C., Vigna, G., Yoshioka, K. and Inoue, D. (2020) Tracing and analyzing web access paths based on {User-Side} data collection: How do users reach malicious {URLs}? In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*: 93–106.

[37] Logger, T. (Accessed 2024), https://github.com/mjuarezm/tablogger.

[38] Duncan, J.F. and Camp, L.J. (2012) Conducting an ethical study of web traffic. In *Proceedings of the 5th USENIX Conference on Cyber Security Experimentation and Test*, CSET'12 (USA): 7.

[39] Alexa (2021), https://s3-us-west-2.amazonaws.com/webcitation/f7333ab1d60b1a81cb2f8f39715bb39ff5228724.

[40] Shao, Y., Hernandez, K., Yang, K., Chan-Tin, E. and Abuhamad, M. (2023) Lightweight and effective website fingerprinting over encrypted dns. In *2023 Silicon Valley Cybersecurity Conference (SVCC)*: 1–8. doi:10.1109/SVCC56964.2023.10165086.

[41] Panchenko, A., Lanze, F., Zinnen, A., Henze, M., Pennekamp, J., Wehrle, K. and Engel, T. (2016) Website fingerprinting at internet scale. In *the 23rd Network and Distributed System Security Symposium (NDSS)*.

[42] Cui, W., Chen, T. and Chan-Tin, E. (2020) More realistic website fingerprinting using deep learning. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)* (IEEE): 333–343.

[43] Zhang, B. and Gearhart, S. (2020) Collecting online survey data: A comparison of data quality among a commercial panel & mturk. *Surv. Pract* **13**.