Investigating Traffic Analysis Attacks on Apple iCloud Private Relay

Ali Zohaib
azohaib@umass.edu
University of Massachusetts Amherst
Amherst, Massachusetts, USA

Jade Sheffey
jsheffey@cs.umass.edu
University of Massachusetts Amherst
Amherst, Massachusetts, USA

Amir Houmansadr
amir@cs.umass.edu
University of Massachusetts Amherst
Amherst, Massachusetts, USA

ABSTRACT

The iCloud Private Relay (PR) is a new feature introduced by Apple in June 2021 that aims to enhance online privacy by protecting a subset of web traffic from both local eavesdroppers and websites that use IP-based tracking. The service is integrated into Apple's latest operating systems and uses a two-hop architecture where a user's web traffic is relayed through two proxies run by disjoint entities.

PR's multi-hop architecture resembles traditional anonymity systems such as Tor and mix networks. Such systems, however, are known to be susceptible to a vulnerability known as *traffic analysis*: an intercepting adversary (e.g., a malicious router) can attempt to compromise the privacy promises of such systems by analyzing characteristics (e.g., packet timings and sizes) of their network traffic. In particular, previous works have widely studied the susceptibility of Tor to website fingerprinting and flow correlation, two major forms of traffic analysis.

In this work, we are the first to investigate the threat of traffic analysis against the recently introduced PR. First, we explore PR's current architecture to establish a comprehensive threat model of traffic analysis attacks against PR. Second, we quantify the potential likelihood of these attacks against PR by evaluating the risks imposed by real-world AS-level adversaries through empirical measurement of Internet routes. Our evaluations show that some autonomous systems are in a particularly strong position to perform traffic analysis on a large fraction of PR traffic. Finally, having demonstrated the potential for these attacks to occur, we evaluate the performance of several flow correlation and website fingerprinting attacks over PR traffic. Our evaluations show that PR is highly vulnerable to state-of-the-art website fingerprinting and flow correlation attacks, with both attacks achieving high success rates. We hope that our study will shed light on the significance of traffic analysis to the current PR deployment, convincing Apple to perform design adjustments to alleviate the risks.

CCS CONCEPTS

• **Security and privacy → Network security**.

KEYWORDS

iCloud Private Relay, Anonymity Systems, Traffic Analysis

ACM Reference Format:
Ali Zohaib, Jade Sheffey, and Amir Houmansadr. 2023. Investigating Traffic Analysis Attacks on Apple iCloud Private Relay. In *ACM ASIA Conference on Computer and Communications Security (ASIA CCS '23), July 10–14, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3579856.3595793

ASIA CCS '23, July 10–14, 2023, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0098-9/23/07…$15.00
https://doi.org/10.1145/3579856.3595793

1 INTRODUCTION

Apple announced iCloud Private Relay (PR) as a new feature for iOS 15, iPadOS 15, and macOS Monterey at WWDC 2021 [1]. Private Relay is part of Apple's privacy-focused recent software releases that allows users with an iCloud+ subscription to browse the Internet through Safari without revealing information about their web traffic, such as IP addresses and DNS queries. The service is available at a moderate base price of $0.99 per month. Apple claims that PR protects users from unwanted tracking by both network providers and website owners, who can use traffic metadata for targeted market campaigns or profile aggregation. It achieves this by ensuring that HTTP traffic leaving an Apple device is encrypted and by routing traffic through exactly two relays controlled by different companies. This guarantees that destination IPs and DNS queries are hidden from the Internet service provider (ISP) and that a user's true IP is hidden from website operators. Apple claims: that "no single entity can combine IP address, location, and browsing activity into detailed profile information" [8].

At first glance, Private Relay comes off as a VPN-like service built with architectural inspiration from The Onion Router (Tor) [22] as well as mix networks; it promises web privacy to the user, is directly integrated into the operating system, and is available at a price less than an average VPN. With Apple's growing global smartphone/laptop market share [44], PR is expected to be used by millions of users. In fact, it has already gained significant traction amongst Internet users [3, 45]. However, partly because of its recent release, there is a lack of in-depth analysis of PR's privacy goals, architecture, and security.

In this paper, we focus on evaluating PR's susceptibility to *traffic analysis* [19, 37, 46]. Traffic analysis is a powerful technique used against anonymity systems that enables an attacker to infer the contents of encrypted traffic (fingerprinting) or deanonymize users of the network (flow correlation). Tor's susceptibility to such attacks has received a great degree of study from the research community, and it is generally accepted that traffic analysis attacks pose a threat to the goals of anonymity systems. For instance, in a *flow correlation* attack [47, 50, 63], an adversary with the ability to passively monitor the ingress and egress traffic of an anonymity

system uses side-channel information such as packet timings, sizes, etc. to correlate traffic and de-anonymize users. Alternatively, *website fingerprinting* attacks can enable attackers to identify a user's browsing activity [10, 51, 59, 60, 66]. PR's vulnerability to any such attacks would weaken Apple's privacy claims and objectives.

Considering the architectural resemblance of PR to systems like Tor, it is reasonable to expect that it may encounter similar problems and be similarly susceptible to traffic analysis attacks. However, because of (1) the different privacy goals of PR and (2) the different architecture of PR, an understanding of the *threat model* is necessary; this is our first goal in this work. In Section 4, we present a comprehensive threat model for traffic analysis adversaries on PR. We list possible scenarios in which an attacker would be enabled to perform flow-correlation and website fingerprinting attacks. We argue that the threat of flow correlation on PR is exacerbated by the design of the system —a two-hop architecture wherein a handful of entities control the full network of relays. We show that autonomous systems (AS) owned by Apple's partners are in a unique position to perform traffic analysis (Section 3.3).

Next, to measure the threat of an AS-level adversary performing traffic analysis on PR, we use algorithmic simulations [26] over the latest Internet Topology graph [27] to predict potentially vulnerable Internet paths. Our analysis in Section 5 finds that 36.8% of connections through the PR are potentially vulnerable against AS-level adversaries, i.e., the presence of an AS on both ingress and egress paths enabling her to perform flow correlation. For instance, PR connections passing through Akamai-controlled AS36183 and AS20940, which contain a large number of PR relays, make up the majority of vulnerable connections. Additionally, we find that PR connections from India, Mexico, and Spain are most susceptible to potential flow correlation attacks, where more than 50% of the possible connections are found to be vulnerable.

Finally, in Section 6 we demonstrate the susceptibility of PR to traffic analysis by performing flow correlation and website fingerprinting (WF) attacks using recent techniques. We use DeepCorr [47] to perform a flow-correlation attack on PR traffic. Moreover, we perform WF attacks on PR using two models: Deep Fingerprinting [59] and Var-CNN [10]. Our results show that PR is susceptible to traffic analysis attacks with both attacks achieving high success rates in different settings.

We believe that our study demonstrates a serious design flaw in PR, rooted in the architectural design and trust assumptions of the system. Although the susceptibility of PR to traffic analysis attacks is not surprising, given that widely-used privacy systems such as Tor are also known to be vulnerable to these attacks, what warrants attention is Apple's architectural choices that exacerbate these threats. We highlight the notion of trust that users place in using PR and advocate for more transparency from Apple about the risks and limitations of such systems. We hope that our work can lead to design adjustments by Apple to alleviate the risks of traffic analysis, such as implementing traffic obfuscation techniques and improving the selection of proxies.

2 BACKGROUND

Anonymous communication and sharing platforms have been around for decades. In 1981, Chaum [16] was the first to introduce the concept of an anonymous email service aimed at concealing the identities involved in an email exchange setting. Since then, there has been a significant development in the domain with applications in problems such as anonymous voting, Private Information Retrieval (PIR), censorship resistance, anonymous web browsing, hidden web services, etc. Tor [22], I2P [73], and Freenet [18] are some of the most well-known and readily available anonymity systems, and works such as MixNets [16], Loopix [53] and Crowds [55] offer alternative perspectives. Approaches to anonymity networks generally involve concepts from peer-to-peer systems, mix networks, onion routing, and private mailbox systems.

Onion Routing is designed to anonymize communication in applications that have low-latency requirements such as web browsing. A typical onion routing network consists of a set of nodes, called *Onion Routers (ORs)*. Users choose a set of ORs to establish a bidirectional channel, popularly known as *circuit* to relay their data through this onion network. As the name suggests, communication in an onion network is encrypted in a layered fashion and each intermediary OR can only decrypt its corresponding layer to forward the data to the next OR. Therefore, only the first OR, or entry node, in the circuit is aware of the IP address of the user who initiated the circuit, and only the last OR, or exit node, is aware of the traffic's destination. In traffic analysis literature, Tor is often targeted because of its privacy guarantees. Additionally, Tor's fixed-sized cells provide some measure of obfuscaion from traffic analysis attacks.

Website Fingerprinting is a traffic analysis technique that allows an attacker to infer the destination of browsing traffic from the metadata available in tunneled or encrypted browsing traffic. Herrmann et al. [33] were first to evaluate Tor against website fingerprinting. Later on, researchers developed techniques that utilized traditional machine learning classifiers that required manual feature engineering to develop a set of best-representative features for a website. Works that build on these techniques include k-NN [66], k-FP [30], and CUMUL [51] and were shown to achieve greater than 90% classification accuracy on closed-world datasets. Other similar works focused on advanced feature analyses to increase the classifier performance [42, 54, 71].

Recent works in website fingerprinting, however, have taken advantage of deep learning mechanisms [56] to strengthen WF attacks and have achieved identification accuracies above 95% in closed-world settings. In 2018, Sirinam et al. proposed Deep Fingerprinting [59], a CNN-based attack that achieved 98.3% accuracy in the closed-world scenario. With Var-CNN [10] and Triplet Fingerprinting [60], researchers introduced sophisticated attacks that were proven to perform better in low-data settings. Smith et al. [61] recently studied the impact of the QUIC protocol on WF attacks, which is also the transport PR uses, though PR uses a different protocol than the standard HTTP-over-QUIC measured by Smith et al. In our work, we perform WF attacks using Deep Fingerprinting and Var-CNN on PR.

Flow Correlation is a method to deanonymize encrypted traffic by correlating ingress and egress traffic to and from an anonymity network, respectively. Similar to WF, Flow Correlation uses side

channel characteristics of network traffic such as packet timings, sizes, and directions to correlate ingress and egress traffic. There are two types of flow correlation attacks - *Active Correlation (Watermarking)* and *Passive Correlation*. In passive correlation attacks, the adversary is assumed to just wiretap both ends of a connection. In active flow correlation, however, an adversary manipulates the features of intercepted traffic to identify the two sides of a flow. Many active flow correlation systems [35, 36, 72] perturb the packet timings of intercepted flows to add delays to the transmission which enables the attacker to generate an artificial pattern for the connection, also known as a watermark. In this paper, we focus on passive flow correlation attacks.

Flow correlation has particularly been studied as an attack on Tor [31, 43, 47, 63]. Traditional methods use statistical metrics such as Pearson Correlation Coefficient, Cosine Similarity, Spearman Correlation rank, etc. to correlate vectors of flow timings and sizes but were found to be inefficient when applied to noisy networks such as Tor. Recent methods use deep learning models to link network flows and have demonstrated high performance. Nasr et al.'s DeepCorr [47] was the first work that used a CNN-based approach to classify network flows on the Tor network with higher accuracy and lower false positive rate. In DeepCoFFEA[50], researchers have used metric learning and amplification to further improve the performance of flow correlation attacks. In this work, however, we have used DeepCorr to perform a flow correlation on PR traffic.

3 UNDERSTANDING HOW PRIVATE RELAY WORKS

PR is built into the networking framework of the Apple operating systems and is available as an add-on feature with an iCloud+ subscription. Users can enable PR from iCloud settings on any Apple device running iOS 15, iPadOS 15, or macOS Monterey or later. Once activated, users can choose one of two strategies that send their location information to PR: 1) "Maintain general location" implies that the egress relay assigned to the user will roughly map to the city the user is actually connecting from. 2) "Use country and time zone" means that an egress relay will be chosen from a broader regional set of IP addresses consistent with the user's country and timezone.

3.1 Architecture

According to Apple, the Private Relay uses an "innovative multi-hop architecture in which users' requests are sent through two separate Internet relays operated by different entities. This way, no single party — including Apple — can view or collect the details of users' browsing activity" [8]. Figure 1 lays out the overall architecture of PR. The following details are based on documentation provided by Apple and our experimental observations.

When PR is enabled, all Safari browsing activity and any HTTP traffic are routed through two relays - **an ingress and an egress relay**. Apple states that the ingress relays are operated by Apple itself while the egress relays are operated by a third-party partner. Though Apple has not explicitly named these trusted partners, a recent study [57] has shown that Akamai, Fastly, and Cloudflare are the operational entities.

Ingress Relays: are the first hop in the two-hop architecture of the PR. Upon visiting a website in Safari, clients first connect to an ingress relay by resolving `mask.icloud.com` via type `A/AAAA` queries using the device's default DNS resolver. An ingress IP is chosen randomly from the response to the DNS query. According to Apple's documentation [8], the first relay "uses a traditional geo-IP lookup" to determine which area is most representative of a user's location. It sends back this information to the client device in the form of a truncated geohash. The client uses this information to initiate a proxy connection to the egress relay via the ingress.

Egress Relays: IP addresses of the egress relays are associated with particular regions or cities and the combination of the ingress and egress proxies is randomly chosen, based on the user's location [9]. Apple provides a public listing of all the active egress IP addresses [4]. The egress IP address corresponds to the actual region or country the user is connecting from. For instance, a request made from San Jose, CA may connect to an egress relay with an IP address corresponding to San Francisco, CA. Egress relays proxy a client's visit to the actual target host.

Onion Encryption: To ensure the separation of information between the proxies, PR uses layered encryption similar to the onion encryption mechanism used in Tor. In a PR connection, proxies decrypt a layer of encryption and pass the data onward to the next hop in the link. In this way, the egress proxy only sees the website accessed and not the user accessing the website while the ingress proxy only sees the client IP and not the website being visited.

Connection Protocols: PR uses proxying technology developed by the Multiplexed Application Substrate over QUIC Encryption (MASQUE) Working Group at the Internet Engineering Task Force (IETF) [8]. Most traffic to ingress relays uses QUIC but the service may fall back to HTTP/2 and TLS in networks where QUIC is blocked or fails. In such cases, instead of resolving `mask.icloud.com` to obtain the ingress relay address, clients resolve `mask-h2.icloud.com` to acquire an ingress IP. The connection to the egress proxy uses HTTP/3 and MASQUE [52, 58]. MASQUE enables the use of QUIC with the ability to combine multiple connections within a single proxy connection. If HTTP/3 is not supported, the connection to the egress proxy falls back to the HTTP CONNECT over TLS method.

To authenticate the proxies, clients validate raw public keys sent [70] in the TLS handshake instead of the usual certificate authentication. Key pinning prevents the use of TLS proxies to intercept and perform man-in-the-middle (MITM) attacks on PR connections, which increases the difficulty of analyzing the underlying protocol.

DNS Resolution: To improve the privacy of DNS requests made while connected to PR, Apple uses Oblivious DNS-over-HTTPS (ODoH) [8, 40]. DNS queries sent through PR are padded and encrypted using Hybrid Public Key Encryption (HPKE) to ensure the first relay cannot look up the domain name the user is visiting. The query is then routed to a DNS-over-HTTPS (DoH) [34] server. The client is able to learn its public IP address subnet from the first relay and includes that information via the query's EDNS0 Client Subnet option [8] to receive optimized responses for its egress IP address. It is important to note, that the initial query made to resolve `mask.icloud.com` is sent unencrypted to the system's default DNS resolver. Subsequent DNS resolutions are, however, resolved via ODoH over PR.

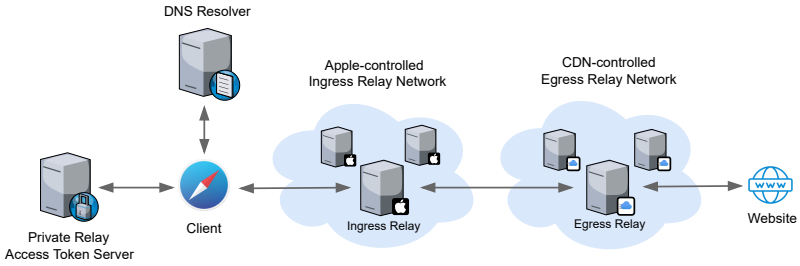

Figure 1: An overview of the Private Relay architecture. ingress nodes are controlled by Apple whereas egress relays are operated by multiple entities.

Client Device Authentication: Apple uses a custom authorization protocol to ensure only legitimate iCloud+ subscribers can connect to PR. According to Apple [8], PR manages network access in a way that does not require identity or account information. This is done using RSA blind signatures to generate anonymous tokens to redeem network access. PR can validate a token using a public key from the access token server. Before, making a connection, the PR access token server provides several tokens to the client device in order to enable it to connect to any proxy operator. These tokens and keys are generated on a daily basis and are rate-limited.

3.2 Comparison with VPNs and Tor

PR provides users with features similar to VPNs and borrows design features from systems such as Tor which makes a case for a comparison between these technologies. It is important to compare these, as traffic analysis attacks that we present in Section 6 have been shown to work against conventional privacy-enhancing technologies such as Tor and certain VPNs. VPNs allow clients to bypass geographic restrictions by routing a user's traffic through a different location and are often used to access content locked in particular regions. PR, however, is designed to comply with geo-blocking and does not hide a user's general location. Additionally, VPN applications usually provide device-wide encryption i.e. all outgoing traffic from the device is encrypted but in the case of PR, only traffic from Safari and HTTP traffic from other applications are sent through PR. Some VPNs offer optional traffic obfuscation [64] to prevent censorship and traffic fingerprinting, a feature notably missing from PR. Other VPNs [2, 7] implement multi-hop support. One key distinction between PR and existing multi-hop VPNs is that existing VPN providers are capable of centrally collecting user data such as browsing history, while PR operators are not independently capable of doing so, due to the promise of separation between Apple and its partners.

In terms of design similarities, the layered encryption mechanism used by PR is very close to that of Tor. As explained previously in Section 3.1, a website request from the client's Safari browser is encrypted in layers with each proxy decrypting its corresponding layer during transmission. While Tor is more strict in terms of privacy, PR shares a geohash of the client's location to the egress relay whereas in Tor's model, the exit relay is only aware of the traffic to and from the destination, and not any information about the client. Unlike Tor, PR does not use any fixed-sized packets and does not employ any obfuscation mechanism. Another major difference between Tor and PR is the number of hops.

Table 1: Comparison between PR, VPNs and Tor

	Private Relay	VPNs	Tor
Hides real IP address	✓	✓	✓
Traffic feature obfuscation	✗	✓	✓
Bypass geo-blocking	✗	✓	✓
Decentralized proxies	✗	✗	✓
Number of hops	2	≥ 1	≥ 3

Tor uses a minimum of 3 hops by default and relays traffic through public relays run by a network of volunteers. On the other hand, PR uses a 2 hop system where the relays are run by major tech companies, with Apple guaranteed to be one of them. While the use of privately controlled relays and two hops gives PR better performance in terms of latency, it is a centralized service, based on trust. This is completely in contrast with the level of trust between Tor users and the relays comprising the Tor network. Table 1 shows a comparison between the three services.

3.3 Current State of the PR Network

In a recent study, Sattler et al. [57] measured the current state of the PR network. They performed a spatial and temporal measurement of the active ingress and egress relays. In their work, they performed DNS queries for `mask.icloud.com` and `mask-h2.icloud.com` over a period of four months to discover 1586 IPv4 addresses and 1575 IPv6 addresses associated with active ingress relays. Based on this discovery, they analyzed ASes that advertise these IP addresses and found that all ingress relay IPs fall under 2 ASes (AS714 and AS36183) which are owned by Apple and Akamai respectively, whereas all egress relay IPs are advertised by 4 ASes (AS36183, AS20940, AS13335, AS54113) which are owned by Akamai, Cloudflare, and Fastly. The presence of Akamai-controlled ASes (AS36183 and AS20940) suggests the possibility of a user's connection going through ASes owned by the same entity, and the researchers reported instances of this in their experiments, although they did not provide information on how often this occurred. To build on these findings, we conduct an experiment to determine how often

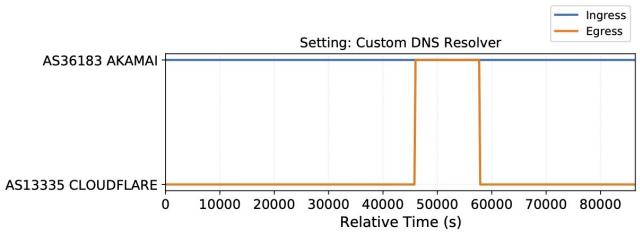

Figure 2: Ingress and Egress IPs mapped to their respective ASes for a two-day measurement where a custom DNS resolver is used to resolve PR domains in a specific AS

ingress and egress relays are located in ASes owned by the same organization and if the selection of ingress relays can be forced via DNS manipulation to affect the subsequent connection routes.

As mentioned previously in Section 3.1, path selection in PR is based on a user's location. Clients typically connect to an ingress relay based on the DNS response they receive from a DNS resolver for `mask.icloud.com` or `mask-h2.icloud.com`. These responses differ based on the user's location, owing to the subnet-based (geolocation-based) response feature in nameservers. Accordingly, setting up a custom DNS resolver should allow a client to connect to a different ingress relay, thereby changing the route of the connection that a user may originally take. Therefore, we use a custom DNS resolver to allow our client to connect to a different but valid ingress IP. Using a local resolver, we set the aforementioned domains to resolve to static ingress IPs that were advertised by AS36183 (i.e. Akamai-owned-AS) and intended for a location that is different from ours. Then, with a laptop running the latest version of macOS and an active iCloud+ account, we use Safari to send HTTP requests to a web server we control. The requests are made every 30 seconds, for a two-day period. To log the ingress IP, we isolate and capture Safari traffic using `tcpdump`. For the corresponding egress IP, we log the source IP of the request on the server. After collecting the ingress and egress IPs, we map IPs to their corresponding ASes and locations.

In our experiment, we make two important observations. First, as shown in Figure 2, for a fixed ingress relay IP in an Akamai AS, Cloudflare was the primary egress AS for the majority of the time. However, there is a 3-hour period where both the ingress and egress relays fell under an Akamai-owned-AS. This not only highlights Akamai's unique position in the PR system but also contradicts Apple's assertion that the ingress and egress relay providers cannot observe both sides of a PR connection. We show that this is not true at the AS level confirming Sattler et al. [57]'s report.

Second, we observe that our client could connect to the fixed ingress relay IP in a different country than ours. This shows that changing the DNS response for PR domains to an ingress IP in another location can actually change the route of the connection that a user may originally take in a normal setting, thereby enabling an adversary to perform a routing attack. Consider a scenario where an AS-level adversary controls or collaborates with an entity that controls any point between the user and the user's DNS server, and wants to place itself on the path between the user and the PR ingress relay. Because the choice of the ingress proxy can be

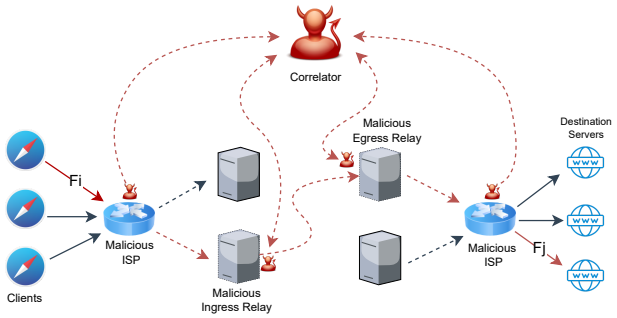

Figure 3: Possible settings for an adversary to perform a flow correlation attack on PR. The adversary can be malicious relays or wiretapping ISPs and/or ASes

manipulated by DNS hijacking, this attacker could modify the DNS response to return a specific ingress relay such that the client is routed through the adversary. A successful routing attack would enable an adversary to intercept web traffic of a user which in turn would enable her to perform traffic analysis attacks. We further explore such attacks in the next section.

4 TRAFFIC ANALYSIS ATTACK MODELS FOR PRIVATE RELAY

According to the developer documentation provided by Apple [5]:

> Private Relay protects users' web browsing in Safari, DNS resolution queries, and insecure HTTP app traffic. Internet connections set up through Private Relay use **anonymous IP addresses** that map to the region a user is in, without divulging the **user's exact location or identity.**

Apple's privacy claims primarily involve hiding traffic from local networks and preventing IP-based tracking from websites. However, there are a number of attacks designed to undermine these protections in systems with even stronger privacy models such as Tor. In this section of the paper, we introduce threat models that could potentially be used to de-anonymize PR users or infer a PR user's browsing traffic.

4.1 Flow Correlation

Flow correlation involves associating ingress flows to an anonymity network with egress flows from the network. Figure 3 shows the possible scenarios where an adversary could perform a flow correlation attack on PR. The goal of an adversary here is to identify the associated flow pairs (F_i, F_j) by comparing the traffic characteristics e.g. packet timings and sizes of all ingress and egress flows. In the case of PR, there are multiple scenarios where an adversary could intercept traffic at various locations to perform a flow correlation attack. Collusion between proxy providers and/or local ISPs and/or website owners can easily enable adversarial attacks to link the ingress and egress flows. Another scenario could involve a local ISP colluding with Apple's partners (who run the egress proxies) to wiretap both sides of the connection and perform a flow correlation attack to de-anonymize PR traffic. Despite Apple's claims that

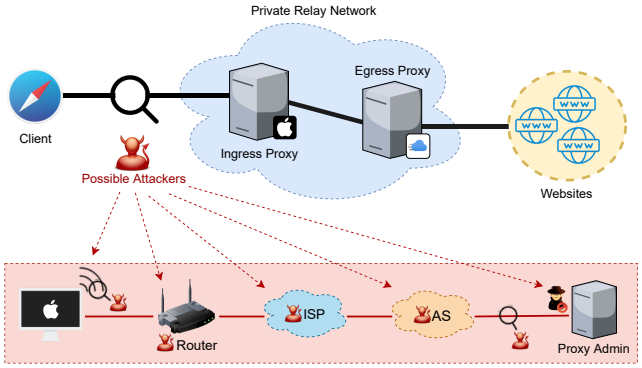

Figure 4: Possible local and passive adversaries for website fingerprinting attacks on Private Relay

the ingress and egress proxies are run by different entities, which provides privacy, they could collude to de-anonymize a client for the purposes of mitigating abuse or cooperating with law enforcement. It is important to note, that in a setting like this, where relay providers are known entities with shared business interests, the notion of privacy is based on trust, which makes a stronger case for a possible flow correlation attack compared to a system like Tor, where trust is not assumed.

In a broader view of a flow-correlation attack, an adversary can increase their chances of performing flow-correlation by controlling or wiretapping autonomous systems (ASes) or Internet Exchange Points (IXPs) and recording transiting traffic. ASes that lie on the path between the source and the ingress and between the egress and the destination are capable of correlating traffic. This case is particularly interesting for PR as all the relays are controlled by a handful of ASes that are capable of monitoring either side of a client's connection. Specifically, the presence of Akamai ASes in both ingress and egress relay paths enhances the possibility of such an attack. In Section 3.3, we show that the case of Akamai ASes controlling both ingress and egress IPs is not uncommon. Additionally, the presence of large ASes on both sides of a PR connection is also possible where the adversarial AS is considered a passive and global eavesdropper. In the case of a successful correlation attack, Apple's claim that no entity would be able to have a full view of a client's connection would be weakened. In Section 5 of this paper, we measure the prevalence of such adversaries and in Section 6 we demonstrate the attack on the active PR network.

4.2 Website Fingerprinting

Website fingerprinting is the practice of identifying a user's web browsing traffic, typically despite some form of obfuscation such as encryption or the traffic shaping applied by an anonymity network. It can be done by network administrators interested in profiling a user for advertisement purposes, censoring access to certain websites, criminal investigations, spying, and any other scenario in which a network-level attacker wants information about a user's browsing traffic. For a website fingerprinting attack, we assume that the adversary seeking to identify the traffic's final destination has access to behavioral metadata such as packet sizes, timings, and directions, but not identifying metadata such as IP addresses or hostnames. Website fingerprinting attacks undermine PR's claim that it protects users from having their traffic monitored by network operators looking to profile their users. Figure 4 shows the possible scenarios and locations where an adversary could intercept PR traffic and perform a website fingerprinting attack. In the case of a local adversary, the attacker is located somewhere between the client and the ingress relay and is assumed to know the identity of a client. They could be an eavesdropper on a client's wireless network, a hacker with access to a user's compromised router, an Internet Service Provider, an AS, or a malicious employee of the ingress proxy provider. In all of these cases, the adversary is assumed to be *passive* implying they only observe traffic flowing through the network and do not manipulate any packets. If PR is vulnerable to website fingerprinting attacks, this casts doubt on Apple's claim that no entity can "connect a user's IP address or account information with their browsing activity." In fact, because Apple happens to be an ingress proxy operator, performing WF on a client allows them to learn the browsing activity of a client.

In Section 6 of this paper, we build on this threat model and perform a website fingerprinting attack on PR.

5 MEASURING AS-LEVEL ADVERSARY PRESENCE

In this section of the paper, we measure the potential presence of the AS-level adversary described in Section 4.1. First, we detail how AS paths between source and destination are determined and explain how potentially adversarial ASes are predicted. Then we explain our experimental methodology and present our results.

5.1 AS-level Path Inference

Identifying paths between arbitrary sources and destinations is necessary to discover the presence of potentially adversarial ASes. Traceroute is a reliable method to discover paths between two ASes, however, finding traceroutes between PR clients, relays, and destinations is unworkable in the case of PR, as only its operators have that level of access. Hence, we rely on inference techniques to predict AS-level paths.

AS-level Topology and Routing: We perform path prediction using an empirically derived AS-level Internet Topology. In this model, the Internet can be viewed as a connected graph. ASes form the nodes and the edges are the links between them. These links are derived from business relationships between ASes. A model introduced by Gao [24] categorizes these relationships into three sorts: customer-to-provider, provider-to-customer and peer-to-peer. A customer gives monetary compensation to a provider for providing bandwidth whereas in peer-to-peer, ASes agree to transit traffic free of cost. The CAIDA [27] team provides the data for the inferred business relationships between ASes that we use to model the AS-graph. The AS-graph can be used to find likely paths between a source and destination AS.

For routing on the AS-graph, we use the popular Gao-Rexford model (GR Routing) [25]. In this model AS paths are based on three order constraints: Local Preference (LP), Shortest Paths (SP), and Tie Break (TB); For LP, paths are ranked based on the next hop where a customer is preferred over a peer and a peer is preferred

over a provider; SP accounts for the shortest paths in locally preferred paths; TB implies the case wherein one of the multiple paths satisfying LP and SP is chosen based on the lowest hash. This model is not without error. In practice, network operators may choose to violate the rules of the model and many other factors may come into play such as the geographic presence of an AS. However, similar to previous work [48], we use GR routing to model AS paths for our objective is to broadly analyze the threat of AS-level attackers. An efficient implementation of GR routing was introduced by Gill et al. in BGPSim [26]. We use a modified implementation of BGPSim, named BGP-SAS [41], that takes as input a set of AS relationships and generates routing trees represented as directed-acyclic-graphs (DAGs). Each tree is a union of all paths found using a modified Breadth First Search (BFS) based on the routing model. For any two ASes, BGP-SAS returns all paths satisfying LP and SP in order.

Identifying Vulnerable Connections: Due to the asymmetric nature of routing on the Internet, AS paths from the client to the server (the forward path), may differ from the return path from the server back to the client. In the standard view of flow correlation attacks, an adversary should either see forward traffic from the client to the ingress together with forward traffic from egress to the server or should see the reverse traffic from the ingress to the client together with the reverse traffic from the destination to the egress. In [63], Sun et al. highlighted the case for an asymmetric attack on Tor traffic, where an adversary that can observe any directional traffic can infer the data flow using TCP Acknowledgement numbers and perform flow correlation. Although most PR connections are made over QUIC, where acknowledgments are encrypted, it falls back to HTTP/2 in the case of a failed QUIC connection. Additionally, an adversary can correlate packet sizes and timings for QUIC connections. Thus we consider the asymmetric case as well.

To formalize the model we consider the following criteria for a PR connection to be vulnerable to a correlation attack: Let $P_{client \to ingress}$ be the set of ASes on the forward path between a client and an ingress proxy including the ingress proxy AS and $P_{ingress \to client}$ be the set of ASes on the reverse path between the ingress and the client. We similarly define the forward and reverse path AS sets between the egress and the destinations as $P_{egress \to dest}$, $P_{dest \to egress}$. We then say a PR connection is vulnerable on the forward path if $P_{client \to ingress} \cap P_{egress \to dest} \neq \emptyset$. And a PR connection is vulnerable on the reverse path if $P_{ingress \to client} \cap P_{dest \to egress} \neq \emptyset$. The combined asymmetric path from client to ingress is then denoted as $P_{ingress \leftrightarrow client} = P_{client \to ingress} \cup P_{ingress \to client}$. And the combined asymmetric path from egress to destination is denoted as $P_{egress \leftrightarrow dest} = P_{egress \to dest} \cup P_{dest \to egress}$ and the asymmetric path from the client to a destination is vulnerable if

$$P_{client \leftrightarrow ingress} \cap P_{egress \leftrightarrow dest} \neq \emptyset$$

Experimental Setup: To assess the potential threat of AS-level adversaries capable of performing flow correlation, we filtered a list of the top 20 countries with the highest number of active ingress relays. This selection was based on the findings of Sattler et al. [57], who reported a significant geographic concentration of relays in North American and European countries. While this analysis could be expanded to all countries, we consider the top 20 countries to be representative of the majority of PR users. To identify these countries, we utilized the most recent ingress relay IP data provided by Sattler et al. and geolocated the IP addresses using the MaxMind Geolite database. The set of 20 countries includes United States (US), Japan (JP), Germany (DE), United Kingdom (UK), Italy (IT), Hong Kong (HK), France (FR), Sweden (SE), Canada (CA), Singapore (SG), Brazil (BR), Spain (ES), Netherlands (NL), Australia (AU), Turkey (TK), India (IN), Mexico (MX), Switzerland (CH), Denmark (DK) and Poland (PL). Next, we select 10 random ASes in each country and send A-type DNS requests to the authoritative nameserver for `mask.icloud.com` with the ECS option carrying a subnet prefix for the selected AS. From the responses, we pick a random IP, and note the corresponding ingress AS via WHOIS information. This is to mimic the ingress selection of a PR client. For the 200 client ASes, we get ingress IPs of which 54% fall in AS36183 and 46% belong to AS714. To find paths between the ASes, we rely on AS-path inference.

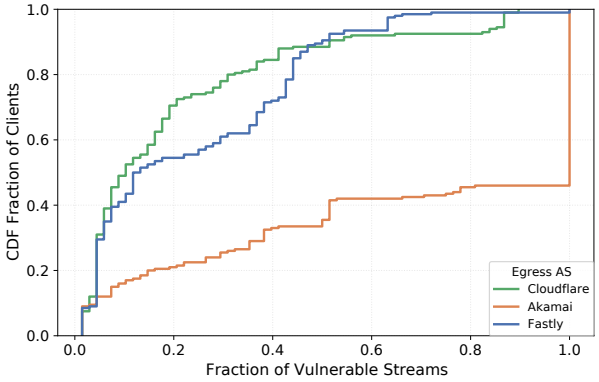

Figure 5: CDF of Client with a Private Relay connection being vulnerable for each Egress AS.

We use AS-path inference using BGP-SAS to obtain all paths satisfying LP and SP between client and ingress ASes. We then filter the top 2 paths (based on the preference order) for each forward and reverse direction. Similarly, we obtain paths between the 3 egress ASes outlined in Section 3.3 (we combine Akamai's sibling ASes as one) and the ASes that advertise the IPs for the top 200 Alexa websites [6]. For each of the client-to-ingress paths, we combine all egress-to-destination paths for each of the egress ASes by taking a cross-product. This gives us a set of all possible paths for a client to a destination via PR. We flag paths as vulnerable based on the adversary model described earlier. This provides an estimate of the threat of network-level attackers on PR. It is important to note that, although Apple claims that PR connections traverse through 2 disjoint operators, we considered all cases, including overlapping ingress and egress AS owners. This is because our observation in Section 3.3 showed the possibility of such an occurrence at the AS level, i.e., the same entity owning ingress and egress proxy ASes.

Results: Of all the routing paths we generate between clients to destinations via PR ASes, 21.4% were found to be vulnerable to at least one network-level attacker on the forward paths while 22.1% were vulnerable to adversaries on the reverse paths. For asymmetric paths, 36.8% of paths were found to be vulnerable.

To better understand the distribution of vulnerable routing paths, Figure 5 shows the estimated distribution of a PR client building a

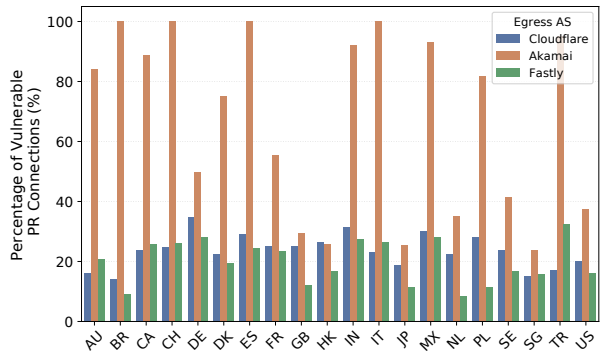

Figure 6: Percentage of vulnerable connections for client ASes per country for each Egress AS.

vulnerable path categorized by the egress AS. We have classified paths by the 3 egress ASes for they are all owned by Apple's partners in comparison to Apple owning one of the two ingress ASes. From our results, it is intuitive to conclude that Akamai's presence in both the sets of ingress and egress ASes puts it in a better position to observe either side of a connection. 42% of the clients making a PR connection via an Akamai egress AS had 80% vulnerable routing paths.

For each of the 20 client countries, we found India (IN), Spain (ES), and Mexico (TK) to be the most vulnerable overall. More than 50% of routing paths originating from each of the 10 ASes in these countries were found to be vulnerable. This can be simply explained by the fact that because clients select the ingress relay by resolving `mask.icloud.com`, regions/subnets where Akamai serves as the only ingress AS, the probability of a vulnerable connection is more as an Akamai AS may also contain the egress relay. Figure 6 shows the percentage of vulnerable paths per country for each egress AS where clients making connections via Akamai egress ASes become more vulnerable. It can be seen that for all the countries, Akamai as an egress AS poses the greatest threat.

In our experiment, we found a total of 68 ASes, with varying percentage presences that were well-positioned to observe both sides of a PR connection as a global passive eavesdropper. Table 2 shows the top 10 ASes found to be intersecting on PR connections. AS20940 and AS36183 exist on 53.2% and 48.9% of the routing paths that were marked vulnerable in our experiment. Both of these ASes are owned by Akamai. Other possible adversaries that are also capable of observing either side of a connection include ASes controlled by large telecom providers such as Vodafone and Sprint. Our main observation in this experiment is that Akamai has a significant presence at the network-level that could enable it to perform a flow correlation attack. The matter most concerning is that this issue is built into the design of PR. Our result confirms and extends the finding of Sattler et al. [57] that traffic can traverse through ASes that are owned by a single entity i.e Akamai. This enhances the threat of traffic analysis on PR.

Table 2: Top 10 ASes and their percentage presence on all vulnerable connections observed in our experiment

ASN	% Presence	AS Name	Country
20940	53.2	Akamai-ASN1	NL
36183	48.9	Akamai-AS	US
4455	14.0	BSO	UK
1273	9.4	Vodafone	UK
1239	9.3	Sprint	US
6939	8.3	Hurricane	US
701	3.4	Verizon/UUnet	US
2516	2.2	KDDI	JP
714	1.8	Apple-Engineering	US
9304	1.7	HGC Global	HK

6 EXPERIMENTAL EVALUATION OF TRAFFIC ANALYSIS ATTACKS

In this section, we elaborate on the various threat models proposed in Section 4, and perform these attacks on PR traffic. First, we explain our experimental and data collection methodology. Then, show our results for the two traffic analysis attacks on PR.[1]

6.1 General Experimental Setup

We design an experimental setup to perform website fingerprinting and flow correlation attacks based on the threat models described in Sections 4.2 and 4.1. For both of these experiments, we used two Macbook M1 machines running macOS Monterey (version 12.2) with 8 Gb of RAM. We used Apple's Open Scripting Architecture (`osascript`) to simulate browsing events, (e.g. full page scrolling) via bash scripts. For each of our experiments, we used Safari version 15.3 and enabled the in-browser option for PR and Cross-site Tracking Prevention. We used a single iCloud+ account and subscription to enable PR on our Macs. We used `tcpdump` to collect the traces and isolate Safari traffic based on process ID. All traces were collected using single browser instances in single-tab sessions.

For our flow correlation experiment, we set up Ubuntu Virtual Machines (VMs), deployed as VPS instances on the Digital Ocean platform. These VMs were set up in different regions and had Apache web servers configured to serve web traffic over HTTP. We collected traffic from two sides for this experiment: one, on our local machines and the second on the servers. Further details about data collection are explained with each attack experiment later in Sections 6.2 and 6.3.

Taking into account ethical considerations, we performed these attacks using only traffic that we generated, and no private data from other PR users was collected or used anywhere in our experiments.

6.2 Flow Correlation attack on PR

As introduced in Section 4.1, Flow correlation is another potentially dangerous attack on anonymity networks. Sometimes referred to as *confirmation attacks*, flow correlation attacks can be used to break anonymity in systems like Tor by correlating traffic features of

[1]Our dataset is available at https://traces.cs.umass.edu/index.php/Network/Network

ingress and egress segments of an encrypted connection [12, 37, 38, 48, 63]. In other use cases, flow correlation can be used to identify malicious users who leverage proxies to hide their identities over the Internet i.e stepping stone attacks. [11, 31, 68]

DeepCorr Classifier: Introduced by Nasr et al [47], DeepCorr is a deep-learning-based classifier that learns a correlation function to associate matching flows on Tor. It was shown to outperform traditional statistical techniques in flow correlation. Although newer models such as DeepCoFFEA [50] have advanced the success of flow-correlation attacks, our aim is not to demonstrate the state-of-the-art of flow correlation, but that these attacks are feasible against PR. We utilized the original implementation of DeepCorr and trained the model on PR traffic collected at two different ends of a connection. Specifically, we trained DeepCorr using 4000 associated flow pairs and used a 1:49 ratio for non-associated flow pairs as that gave the best results for PR data. For comparison between Tor and PR traffic, we used the dataset collected by Nasr et al. [47] for Tor and trained two models, one with PR data and one with Tor flows. Equal-sized training sets were used for both models with the same non-associated flow pairs ratio. We tested each on 500 additional associated and 500x49 non-associated flows.

Data Collection and Features: Using our experimental setup, we set up 5 Linux Virtual Machines on the Digital Ocean cloud service in various locations including the United States, Netherlands, and India. We then manually selected and cloned 100 web pages from a variety of content-heavy websites and modified them to embed a random number of gif images on every page. This was to ensure variance in the traces and that each flow contains at least 300 packets. We uploaded each of these 100 modified webpages to our 5 servers. We then navigated to each of these websites and collected traces on both ends of the connection. In total, we collected 4500 flows over a two-week period. In order to train DeepCorr, we extracted inter-packet delays and packet sizes from the collected traces. We chose flow lengths of 100 and 350 packets to show two data settings and the impact of increasing packets per flow.

Limitations: Collecting real-world data for flow correlation becomes non-trivial without having control over the destination websites/servers. In prior works, researchers have used other techniques such as tunneling exit traffic through a controlled proxy to capture traffic [47]. This, however, is infeasible in the case of PR due to the closed-source nature of the service and the fact that PR prevents the use of nested proxies. Hence, we deploy our own web servers to enable us to collect traffic at both ends of a PR connection. While we acknowledge the setup of our own web servers and websites may not fully represent a real-world setting, similar techniques have been used in previous works [63] to generate traffic via a live anonymity network.

Evaluation Metrics and Results: Similar to previous work [47], we use *true positive (TP)* and *false positive (FP)* error rates as the main metrics to evaluate the performance of flow correlation on PR. TPR measures the fraction of associated pairs of traces that are correctly classified to be correlated whereas the FPR measures the number of non-associated flows incorrectly classified as correlated by DeepCorr. Since the detection threshold makes a trade-off between the TP and FP, we also use the ROC curve to show the performance of the classifier.

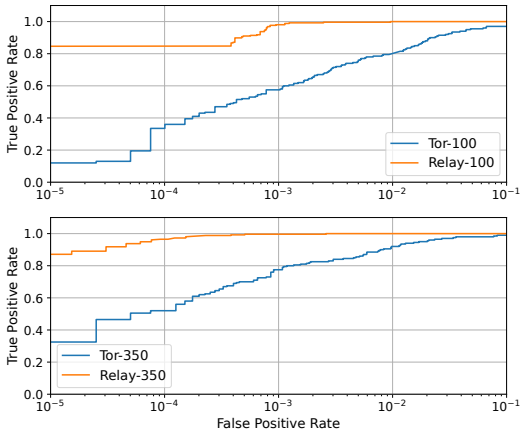

Figure 7: Comparison of DeepCorr on Tor and Private Relay Flows for different flow sizes: 100 and 350 packets

Figure 7 shows the ROC curves of DeepCorr trained on Tor and PR traffic with flow sizes of 100 and 350 packets. We can observe that DeepCorr performs much better on PR flows compared to Tor flows. Intuitively, this follows in line with the fact that PR is a newer network while Tor is overloaded, and has fixed-sized cells, which makes Tor traffic much noisier than PR traffic. At a false positive rate of 10^{-3}, DeepCorr achieves a TPR of 0.8 on Tor traffic with a flow size of 350 whereas, on PR, it achieves a perfect TPR of 1.0. For a flow size of 100 packets, DeepCorr achieves a 0.97 TPR on PR data as compared to the 0.6 TPR on Tor data. Our results also match the observations of Nasr et al. [47] for their stepping stone experiments on CAIDA traces [15] as those traces have a similar low-noise setting to our PR data. Although unsurprising, our results show that PR is susceptible to a flow-correlation attack where an adversary recording traffic metadata from two sides of a PR connection can effectively perform correlation to de-anonymize a PR user.

6.3 Website Fingerprinting Attack on PR

Website fingerprinting is a widely-studied attack on anonymity systems. As introduced in Section 4, in a WF attack, an attacker attempts to identify the website visited by a user by comparing it to previously collected traces of known websites. Because the encapsulating protocol is encrypted, the attacker utilizes side-channel information from the packets such as timestamps and payload sizes to train a classifier.

WF most often uses two settings to evaluate WF models - *closed-world* and *open-world*. In the closed-world setting, all traces are known to be from a monitored set of N web pages. The classifier, having been trained on sample traffic from pages in that set determines which of the N web pages the trace corresponds to. In the open-world setting, the feature vector presented can either belong to the set of N monitored or unmonitored web pages. The objective, in this case, is to identify if the trace belongs to a monitored web page and if so, which, or whether it is of some unmonitored web page. Many WF techniques [10, 49, 56] in the literature use a variety of assumptions about web browsing that simplify and enhance the

performance of the classifiers. These assumptions are largely based on synthetic traffic generation [17] and include the adversaries' ability to detect the beginning and end of a web page load, the absence of background traffic in a visit, and classification only based on the index page of the website. We note that PR relays are solely operated by Apple/partners, so one can not collect PR traffic of actual users. We, therefore, resort to using our own synthetic traffic for the purpose of experiments and inherit the aforementioned assumptions in our experiments.

We consider both the closed and open-world settings for our experiments. While sizes of the open-world datasets have increased dramatically in recent times, researchers still rely on datasets of 4k-20k [49, 60, 65] traces to train and evaluate their classifiers. We have used a similar-sized dataset in this work.

Collecting Traces: For the closed world setting, we sampled 100 websites of the Alexa Top 1M websites [6], accounting for failures such as IP blocking or timeouts. We also ignored duplicate entries of top-level domains which could result in bias due to the shared infrastructure. For example, only one of `google.com` and `google.co.jp` was included in the monitored list. Then, for each of these 100 monitored websites, we browsed the homepages 250 times each to ensure a final quota of 200 instances per website (monitored set). For the open-world setting, we collected one instance each of an additional 20000 websites randomly sampled from the top 1M Alexa websites [6] (unmonitored set), excluding the sites used for the closed-world set.

We collected the traces using our experimental setup described in Section 6.1 through our campus network. For half of the traces, the "Use General Location" option in the PR settings was selected and for the other half "Country and Time zone" option was enabled. We followed data collection methodologies derived from prior work [59] and visited websites in chunks of 25 visits per website. All websites were visited sequentially for 20 seconds each with PR and cross-website tracking protection option enabled in the browser. We introduced a delay of 15 seconds after opening Safari before navigating to each website in order to avoid browser bootstrap traffic.

Website Fingerprinting Classifiers: Previous works in the WF domain have proposed numerous models based on various techniques. We, however, use two of these models to represent effective WF techniques. The following are the classifiers used:

Deep Fingerprinting Model: Based on convolutional neural networks (CNN), Deep Fingerprinting (DF) is the deep-learning classifier proposed by Sirinam et al [59]. DF uses techniques from computer vision to extract features from sequences of packets for classification. DF was shown to perform well against popular WF defenses such as WTF-PAD [39] and Walkie-Talkie [67].

Var-CNN: The Var-CNN classifier introduced by Bhat et al. [10] is another deep-learning classifier we evaluate in this work. It uses automatic and manual feature detection to train two Residual Networks (ResNets) on packet directions and timings and was shown to perform much better than other classifiers on smaller datasets.

Borrowing ideas from Smith et al. [61], we included the packet sizes along with the packet direction to train the Deep Fingerprinting [59] and Var-CNN [10] models. Unlike the Tor setting used in most of the previous works, where packets are padded to be of fixed length and are shown to not contribute to classifier performance,

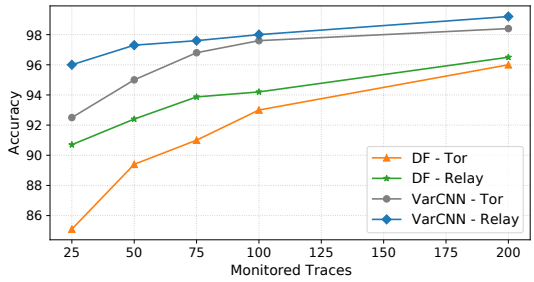

Figure 8: Closed-World Accuracy as a function of the number of monitored traces per website for VarCNN and Deep Fingerprinting Models on PR and Tor traffic

packet sizes are available in the PR setting and hence we provide the two classifiers with signed packet sizes instead of just packet directions. Similar to the original Var-CNN's packet count features, we also provided Var-CNN with additional aggregate packet size features including incoming, outgoing, total bytes, and the ratio of incoming and outgoing bytes.

Evaluation: In the closed-world setting, a finite set of monitored web pages is used and the attacker is assumed to know this set and therefore train the classifier on it. Similar to previous works, we have used `accuracy` as the metric of evaluation in this scenario. Accuracy is defined as the ratio of the number of correctly classified traces to the total number of traces in the closed-world dataset.

In the closed-world setting consisting of 100 websites with 200 instances of each, DF gave an accuracy of **96.5%** whereas Var-CNN was able to achieve an accuracy of **99.2%**. For comparison, we used the Tor dataset from Rimmer et al. [56] and trained both models over both datasets in similar settings. Figure 8 compares the closed-world accuracies of both models trained on different datasets for Tor and PR. We observe that both classifiers perform better on PR traffic than on Tor traffic.

In the open-world scenario, the user is assumed to visit any website in the world. For this setting, we train a classifier with 100 instances of 100 monitored set websites along with one instance of another 9000 unmonitored set websites. Our test set consists of 20000 instances containing an equal number of traces from our monitored and unmonitored sets. Similar to previous works [59], we evaluate the performance of the classifier in this scenario using True Positive Rate (TPR), False Positive Rate (FPR), Precision, and Recall. Our attack with the DF model achieves a True Positive Rate of 0.995 and False Positive Rate of 0.007 when optimizing for high TPR and achieves a TPR of 0.88 and FPR of 0.0002 when optimizing for low FPR. Figure 9 shows the ROC and Precision/Recall curve for our experiment with the DF model in the open-world setting.

Our results are in line with our intuitive understanding of PR: because PR does not employ an obfuscation mechanism to hide its traffic characteristics, it is susceptible to traffic analysis attacks. Obfuscation and other countermeasures that are deployed for systems like Tor hamper the performance of WF classifiers. The absence of any significant defense against traffic analysis attacks makes PR susceptible to website fingerprinting.

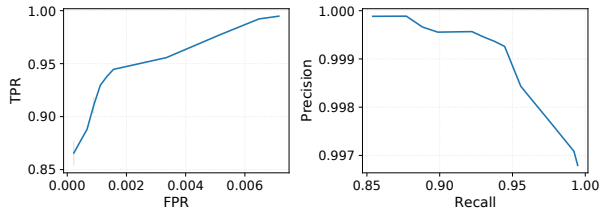

Figure 9: ROC and Precision/Recall curve for Open-World Evaluation using Deep Fingerprinting model on PR Traffic

7 DISCUSSIONS

Traffic analysis techniques are known to work against conventional privacy-enhancing technologies such as Tor. Because PR has significant architectural similarities with Tor except that has no features that would obfuscate traffic characteristics, it is *not* surprising that PR is also vulnerable to traffic analysis. In fact, Section 8 of the MASQUE [52] draft notes the susceptibility of the protocol to flow correlation.

Given that Apple was directly involved in the development of MASQUE, it is reasonable to argue that Apple is aware of the threat of traffic analysis but has made architectural choices based on a tradeoff between trust, performance and privacy. PR's design provides significant benefits in terms of both performance and usability. Additionally, given that all PR traffic is processed a relatively small number of high-bandwidth servers, the sheer volume of PR traffic could increase the difficulty of a flow correlation attack, if collusion were to occur. Despite the benefits, the privacy guarantees in PR are reliant on the assumption of non-collusion. This warrants attention from its users and transparency its providers. We argue that compared to a decentralized trust model where performing a flow correlation attack requires acquiring control of a large portion of the network, compromising user privacy in PR is far more feasible in the first place, as it only requires collusion between the few operational entities.

This raises the question of what possible improvements can be made to PR to minimize the threat of traffic analysis, which could enhance the privacy of millions of users. Traffic analysis attacks generally introduce a performance overhead, but Apple could compromise by adopting an opt-in approach for users with stronger threat models. This has some precedent: Apple offers a "High Security Mode" [69] for users at risk of targeted zero-day exploits.

As a first step against flow correlation threats, Apple should ensure that PR never uses the same provider for both the ingress and egress portion of a PR connection, as shown by our experiments in Section 3.3. Next, AS-aware path selection algorithms can be designed to prevent flow correlation attacks from entities outside of Apple and their partners. Similar systems proposed for Tor include Counter-RAPTOR [62] and DPSelect [29]. Given that PR is also used in Apple's mobile devices with multiple potential internet connections, multihoming [32] has been shown to provide benefits against traffic analysis attacks, though it requires a specific network environment and may incur charges. To mitigate the effect of a single ingress or path on the potential to perform flow correlation against PR users, TrafficSliver [21] or MPTCP [20] could be applied. A number of defenses have been proposed against WF attacks [13, 14, 23, 28, 66, 67]. Works such as BuFLO [23], CS-BuFLo [13] and Tamaraw [14] use rate fixing and padding to conceal traffic features but incur significant overhead. To tackle the issue of high overhead, lightweight countermeasures such Walkie-Talkie [67] and WTF-PAD [39] were proposed for deployment in Tor. While we acknowledge the performance/safety trade-off PR makes, making any of these mitigations available as an optional feature would enhance overall user privacy.

8 CONCLUSION

In this work, we analyzed the threat of traffic analysis against Apple's new privacy tool — iCloud Private Relay. We presented a comprehensive threat model for PR that outlines how PR users could become vulnerable to traffic analysis attacks. We showed a scenario where a client's traffic can traverse through relay ASes that are controlled by the same entity which contradicts Apple's claims and amplifies the possibility of traffic analysis. We measured the risk imposed by AS-level adversaries through empirical measurement of Internet routes. To demonstrate the susceptibility of PR against traffic analysis attacks, we performed website fingerprinting and flow correlation attacks on PR traffic and showed that PR is susceptible to these attacks. We hope our work will aid discussion and create awareness about the traffic analysis threats on PR and encourage Apple to deploy countermeasures against these attacks as optional features.

ACKNOWLEDGMENTS

This work was supported in part by the NSF grant CNS-1953786, by the Young Faculty Award program of the Defense Advanced Research Projects Agency (DARPA) under the grant DARPA-RA-21-03-09-YFA9-FP-003, and by DARPA under Agreement No. HR00112190125. The views, opinions, and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for public release; distribution is unlimited.

REFERENCES

[1] 2022. About iCloud Private Relay. https://support.apple.com/en-us/HT212614
[2] 2022. Extra Security With Double VPN | NordVPN. https://nordvpn.com/features/double-vpn/. (Accessed on 04/19/2023).
[3] 2022. Immue discovers new exploitation of Apple's private relay | VentureBeat. https://venturebeat.com/security/immue-discovers-new-vulnerability-in-apples-private-relay/. (Accessed on 08/30/2022).
[4] 2022. *List of Private Relay Egress IPs.* https://mask-api.icloud.com/egress-ip-ranges.csv
[5] 2022. Prepare Your Network or Web Server for iCloud Private Relay - Support - Apple Developer. https://developer.apple.com/support/prepare-your-network-for-icloud-private-relay/. (Accessed on 08/25/2022).
[6] 2023. Alexa Top Sites. http://s3.amazonaws.com/alexa-static/top-1m.csv.zip.
[7] 2023. Multihop with WireGuard - Guides | Mullvad VPN. https://mullvad.net/en/help/multihop-wireguard/. (Accessed on 04/19/2023).
[8] Apple. 2021. *iCloud Private Relay Overview.* https://www.apple.com/privacy/docs/iCloud_Private_Relay_Overview_Dec2021.PDF
[9] Apple. 2021. WWDC 2021 - Video. (2021). https://developer.apple.com/videos/play/wwdc2021/10085/
[10] Sanjit Bhat, David Lu, Albert Hyukjae Kwon, and Srinivas Devadas. 2019. Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. *PETS 2019* (2019).
[11] Avrim Blum, Dawn Song, and Shobha Venkataraman. 2004. Detection of interactive stepping stones: Algorithms and confidence bounds. In *International Workshop on Recent Advances in Intrusion Detection*. Springer.

[12] Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. 2007. Denial of service or denial of security?. In *ACM CCS 2007*.

[13] Xiang Cai, Rishab Nithyanand, and Rob Johnson. 2014. CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense. In *WPES 2014*.

[14] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. 2014. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *ACM CCS 2014*.

[15] CAIDA. 2016. Anonymized Internet Traces 2016. https://catalog.caida.org/dataset/passive_2016_pcap. (Accessed on 08/29/2022).

[16] David L Chaum. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM* (1981).

[17] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. 2022. Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World. In *USENIX Security 2022*.

[18] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W Hong. 2001. Freenet: A distributed anonymous information storage and retrieval system. In *Designing privacy enhancing technologies*. Springer.

[19] George Danezis. 2004. The traffic analysis of continuous-time mixes. In *International Workshop on Privacy Enhancing Technologies*. Springer.

[20] Wladimir De la Cadena, Daniel Kaiser, Andriy Panchenko, and Thomas Engel. 2020. Out-of-the-box Multipath TCP as a Tor Transport Protocol: Performance and Privacy Implications. In *IEEE NCA 2020*.

[21] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. 2020. TrafficSliver: Fighting Website Fingerprinting Attacks with Traffic Splitting. In *ACM CCS 2020*.

[22] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *USENIX Security 2004*.

[23] Kevin P Dyer, Scott E Coull, Thomas Ristenpart, and Thomas Shrimpton. 2012. Peek-a-boo, I still see you: Why efficient traffic analysis countermeasures fail. In *IEEE S&P 2012*.

[24] Lixin Gao. 2001. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking* (2001).

[25] Lixin Gao and Jennifer Rexford. 2001. Stable Internet routing without global coordination. *IEEE/ACM Transactions on networking* (2001).

[26] Phillipa Gill, Michael Schapira, and Sharon Goldberg. 2012. Modeling on quicksand: dealing with the scarcity of ground truth in interdomain routing data. *ACM SIGCOMM CCR* (2012).

[27] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and KC Claffy. 2014. Inferring complex AS relationships. In *IMC 2014*.

[28] Jiajun Gong and Tao Wang. 2020. Zero-delay lightweight defenses against website fingerprinting. In *USENIX Security 2020*.

[29] Hans Hanley, Yixin Sun, Sameer Wagh, and Prateek Mittal. 2019. DPSelect: a differential privacy based guard relay selection algorithm for Tor. *PETS 2019* (2019).

[30] Jamie Hayes and George Danezis. 2016. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *USENIX Security 2016*.

[31] Ting He and Lang Tong. 2007. Detecting encrypted stepping-stone connections. *IEEE Transactions on Signal Processing* (2007).

[32] Sébastien Henri, Gines Garcia-Aviles, Pablo Serrano, Albert Banchs, and Patrick Thiran. 2020. Protecting against Website Fingerprinting with Multihoming. *PETS 2020* (2020).

[33] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. 2009. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *CCSW 2009*.

[34] Paul E. Hoffman and Patrick McManus. 2018. DNS Queries over HTTPS (DoH). RFC 8484. https://doi.org/10.17487/RFC8484

[35] Amir Houmansadr and Nikita Borisov. 2011. SWIRL: A Scalable Watermark to Detect Correlated Network Flows. In *NDSS 2011*.

[36] Amir Houmansadr and Nikita Borisov. 2011. Towards improving network flow watermarks using the repeat-accumulate codes. In *IEEE ICASSP 2011*.

[37] Rob Jansen, Marc Juárez, Rafa Galvez, Tariq Elahi, and Claudia Diaz. 2018. Inside Job: Applying Traffic Analysis to Measure Tor from Within. In *NDSS 2018*.

[38] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. 2013. Users get routed: Traffic correlation on Tor by realistic adversaries. In *ACM CCS 2013*.

[39] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. 2016. Toward an Efficient Website Fingerprinting Defense. In *ESORICS 2016*.

[40] Eric Kinnear, Patrick McManus, Tommy Pauly, Tanya Verma, and Christopher A. Wood. 2022. Oblivious DNS over HTTPS. RFC 9230. https://doi.org/10.17487/RFC9230

[41] Kirtus G Leyba, Benjamin Edwards, Cynthia Freeman, Jedidiah R Crandall, and Stephanie Forrest. 2019. Borders and Gateways: Measuring and Analyzing National as Chokepoints. In *ACM COMPASS 2019*.

[42] Shuai Li, Huajun Guo, and Nicholas Hopper. 2018. Measuring information leakage in website fingerprinting attacks and defenses. In *ACM CCS 2018*.

[43] Zhen Ling, Junzhou Luo, Wei Yu, Xinwen Fu, Dong Xuan, and Weijia Jia. 2009. A new cell counter based attack against tor. In *ACM CCS 2009*.

[44] Ben Lovejoy. 2022. iPhone US market share hits all-time high, overtaking Android. https://9to5mac.com/2022/09/02/iphone-us-market-share/. (Accessed on 12/15/2022).

[45] Sergey Mostsevenko. 2021. iCloud Private Relay Vulnerability Identified. https://fingerprintjs.com/blog/ios15-icloud-private-relay-vulnerability/

[46] S.J. Murdoch and G. Danezis. 2005. Low-cost traffic analysis of Tor. In *IEEE S&P 2005*.

[47] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. 2018. DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In *ACM CCS 2018*.

[48] Rishab Nithyanand, Oleksii Starov, Adva Zair, Phillipa Gill, and Michael Schapira. 2016. Measuring and Mitigating AS-level Adversaries Against Tor. In *NDSS 2016*.

[49] Se Oh, Saikrishna Sunkam, and Nicholas Hopper. 2019. p1-FP: Extraction, Classification, and Prediction of Website Fingerprints with Deep Learning. *PETS 2019* 2019 (2019).

[50] Se Eun Oh, Taiji Yang, Nate Mathews, James K Holland, Mohammad Saidur Rahman, Nicholas Hopper, and Matthew Wright. 2022. DeepCoFFEA: Improved Flow Correlation Attacks on Tor via Metric Learning and Amplification. In *IEEE S&P 2022*.

[51] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. 2016. Website Fingerprinting at Internet Scale. In *NDSS 2016*.

[52] Tommy Pauly, Eric Rosenberg, and David Schinazi. 2023. *QUIC-Aware Proxying Using HTTP*. Internet-Draft draft-pauly-masque-quic-proxy-06. IETF. https://datatracker.ietf.org/doc/draft-pauly-masque-quic-proxy/06/ Work in Progress.

[53] Ania M Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis. 2017. The loopix anonymity system. In *USENIX Security 2017*.

[54] Mohammad Saidur Rahman, Payap Sirinam, Nate Mathews, Kantha Girish Gangadhara, and Matthew Wright. 2020. Tik-Tok: The Utility of Packet Timing in Website Fingerprinting Attacks. *PETS 2020* (2020).

[55] Michael K Reiter and Aviel D Rubin. 1998. Crowds: Anonymity for web transactions. *ACM TISSEC* (1998).

[56] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *NDSS*.

[57] Patrick Sattler, Juliane Aulbach, Johannes Zirngibl, and Georg Carle. 2022. Towards a tectonic traffic shift?. In *ACM IMC 2022*.

[58] David Schinazi. 2022. Proxying UDP in HTTP. RFC 9298. https://doi.org/10.17487/RFC9298

[59] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *ACM CCS 2018*.

[60] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. 2019. Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In *ACM CCS 2019*.

[61] Jean-Pierre Smith, Prateek Mittal, and Adrian Perrig. 2021. Website Fingerprinting in the Age of QUIC. In *PETS 2021*.

[62] Yixin Sun, Anne Edmundson, Nick Feamster, Mung Chiang, and Prateek Mittal. 2017. Counter-RAPTOR: Safeguarding Tor against active routing attacks. In *IEEE S&P 2017*.

[63] Yixin Sun, Anne Edmundson, Laurent Vanbever, Oscar Li, Jennifer Rexford, Mung Chiang, and Prateek Mittal. 2015. RAPTOR: Routing attacks on privacy in Tor. In *USENIX Security 2015*.

[64] TunnelBear. 2021. TunnelBear implements OpenVPN3 with Pluggable Transports. https://www.tunnelbear.com/blog/tunnelbear-implements-pluggable-transports-with-openvpn3/. (Accessed on 04/19/2023).

[65] Tao Wang. 2020. High Precision Open-World Website Fingerprinting. In *IEEE S&P 2020*.

[66] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. 2014. Effective attacks and provable defenses for website fingerprinting. In *USENIX Security 2014*.

[67] Tao Wang and Ian Goldberg. 2017. Walkie-Talkie: An Efficient Defense Against Passive Website Fingerprinting Attacks. In *USENIX Security 2017*.

[68] Xinyuan Wang and Douglas S Reeves. 2003. Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. In *ACM CCS 2023*.

[69] Zack Whittaker. 2022. Apple says Lockdown Mode in iOS 16 will help block government spyware attacks | TechCrunch. https://techcrunch.com/2022/07/06/apple-lockdown-mode/. (Accessed on 04/20/2023).

[70] Paul Wouters, Hannes Tschofenig, John IETF Gilmore, Samuel Weiler, and Tero Kivinen. 2014. Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS). RFC 7250. https://doi.org/10.17487/RFC7250

[71] Junhua Yan and Jasleen Kaur. 2018. Feature Selection for Website Fingerprinting. In *PETS 2018*.

[72] Wei Yu, Xinwen Fu, Steve Graham, Dong Xuan, and Wei Zhao. 2007. DSSS-based flow marking technique for invisible traceback. In *IEEE S&P 2007*.

[73] Bassam Zantout, Ramzi Haraty, et al. 2011. I2P data communication system. In *ICN 2011*.