Exposing and Mitigating Website Fingerprinting Threats in Named Data Networking

Michael Wrana
University of Waterloo
mmwrana@uwaterloo.ca

Anhelina Bodak
Lviv Polytechnic National University
anhelina.bodak.kb.2022@lpnu.ua

Ansh Dhingra
University of Waterloo
a24dhingra@uwaterloo.ca

Diogo Barradas
University of Waterloo
diogo.barradas@uwaterloo.ca

Abstract

Website fingerprinting (WF) attacks exploit traffic metadata to infer users' web activity, even when content is encrypted. While extensively studied in TCP/IP networks, the feasibility of WF attacks in content-centric architectures such as Named Data Networking (NDN) remains unclear. Unlike TCP/IP, NDN retrieves named content through interest-data exchanges and extensive router caching, reshaping how network metadata manifests. This paper presents the first large-scale empirical study of WF in NDN, moving beyond prior theoretical discussions to systematically evaluate how WF attacks and defenses behave in this different architectural setting.

To enable this analysis, we develop a modular testbed that enables realistic NDN-based experimentation, including realistic web access over ANDaNA, an NDN-native privacy-enhancing technology. We show that state-of-the-art single- and multi-tab deep learning WF attacks remain effective against ANDaNA in both close- and open-world settings, while existing WF defenses (originally built for Tor in TCP/IP networks) are incompatible with NDN's architecture when ported. We conclude by adapting four padding-based WF defenses to NDN's content-centric communication model, improving resilience against WF attacks by up to 80%.

Keywords

future Internet architectures, privacy-enhancing technologies, surveillance, traffic analysis, website fingerprinting

1 Introduction

Future Internet Architectures (FIAs) aim to address long-standing scalability, efficiency, and security limitations of TCP/IP [17]. One major direction within this space is the development of Content-Centric Networks (CCNs), which replace host to host communication with content based retrieval [6]. In CCNs, users request named content rather than connect to specific servers, and those requests can be satisfied by any server or cache with a copy.

One prominent CCN-based architecture is Named Data Networking (NDN) [111], which leverages in-network caching and name-based routing to enable scalable, efficient, and decentralized content delivery. NDN has already been deployed as part of the globally distributed NDN6 testbed, which interconnects dozens of sites across North America, Europe, and Asia [45]. This long-running platform provides sustained operational experience in NDN for edge computing [26], Internet-of-Things [39], intelligent transportation [5], and environmental monitoring [40]. In parallel, researchers have designed NDN-based alternatives to web-based applications like GitHub or Overleaf [110], and explored how NDN can run alongside HTTP(S) [94]. Altogether, with continued development of NDN-focused in-kernel protocol stacks [58], support from programmable data planes [86, 92], and early signs of scalability in testbed deployments [96], it is conceivable that NDN could be phased into broader real-world use, either incrementally alongside IP [16, 114] (via tunnelling) or directly within targeted domains such as smart cities [18] and national backbone infrastructures [4].

Alongside the global NDN6 testbed, experimental NDN deployments spanning seven major metropolitan regions in China [59] have begun to probe NDN's behavior in environments defined by pervasive traffic inspection and information-control mechanisms [104]. This trajectory is poised to bring NDN into settings where online surveillance and censorship are actively exercised, pointing towards a pressing need to examine how adversaries can monitor or suppress user activity. Waiting for broader adoption before scrutinizing NDN's privacy properties risks repeating a common pattern in networking research, where design choices solidify before their security consequences are fully understood [3].

Existing work has already shown that NDN's shift from endpoint-based communication to content-based retrieval introduces unique privacy risks that differ from TCP/IP [103]. For instance, in NDN, a network adversary can infer users' web browsing activities by analyzing content names embedded into packet headers [9], enabling user profiling and/or censorship [64]. Ensuring that content can be retrieved privately is thus a prerequisite for safe adoption of NDN.

To protect users from state-level censorship and profiling, several Privacy-Enhancing Technologies (PETs) were proposed for NDN. Anonymity systems such as ANDaNA [22] and AC3N [95] make use of layered encryption and onion routing principles (similarly to Tor) to hide content names and paths. Other technologies focus on content name obfuscation [64], anonymity through distributed servers [7], and refraction networking [105] to thwart surveillance. While these technologies offer promising privacy guarantees (at least in theory), Wrana et al. [103] have recently challenged the assumption that such protections are sufficient in practice. As it stands, little is known about the resilience of these PETs against

Proceedings on Privacy Enhancing Technologies YYYY(X), 1–17

https://doi.org/XXXXXXX.XXXXXXX

adversaries capable of sophisticated traffic analysis attacks [97] with the goal of disclosing users' encrypted content accesses [79].

Among the array of different encrypted traffic analysis attacks [25, 73, 98], website fingerprinting (WF) [21] stands out as one of the most significant threats to PETs like onion routing systems, virtual private networks (VPNs) [108] or anonymity networks [21]. WF attacks enable a network eavesdropper to identify the specific websites a target visits by analyzing their encrypted traffic metadata (typically packet timing/direction). State-of-the-art WF techniques analyze traffic metadata characteristics using deep learning (DL) to identify visited websites [78, 84] with high accuracy even when observing only a fraction of a page load [21].

The intersection of NDN and WF introduces a previously unexplored security problem. Even though content names are hidden, an NDN website load still leaves metadata that may reveal which resources the browser retrieved. NDN fundamentally changes how that sequence is produced compared with TCP/IP. The user sends interests for named content, data can be returned by any network cache, and every returned data packet depends on forwarding state established by previous interests. Consequently, the portability of WF attacks and the portability of their defenses must be evaluated.

This paper presents the first analysis of WF attacks and defenses in NDN. While prior work has raised theoretical concerns about NDN's susceptibility to traffic analysis [103], we move beyond speculation by conducting an empirical study of how WF manifests in content-centric architectures. To support this effort, we build a modular platform for large-scale NDN experimentation that modernizes legacy NDN tooling and adds support for key application-layer functionalities which were previously limited or absent. Using this platform, we generate the first dataset for WF analysis in NDN, comprising 80,000 page loads of 10,100 unique webpages across NDN deployments on a 257-node RocketFuel [91]-derived ISP topology. We publicly release our dataset and testbed [102].

We use our simulation platform to evaluate WF attacks and defenses against ANDaNA, an NDN-based PET that implements onion routing. ANDaNA's design, which follows an approach similar to Tor's, makes it a natural NDN-native PET for studying whether established WF attacks and defenses transfer to NDN. We show that NDN does not prevent prominent DL WF attacks spanning single- (e.g., Tik-Tok [78], RF [84]) and multi-tab (e.g., TMWF [43], ARES [19]) settings from achieving high accuracy. We characterize the semantic incompatibilities that arise when TCP/IP-oriented defenses are ported to NDN, then adapt four established WF defenses into content-centric (CC) variants. We find that NDN's stateful interest–data semantics constrain the defensive design space, but our CC-defense variants showcase one way to meet this requirement while providing effective countermeasures to WF attacks.

2 Building Towards Traffic Analysis in NDN

2.1 The NDN Architecture and Simulation

This section overviews the core principles of the NDN architecture (see [39]) and surveys existing NDN simulation frameworks.

NDN principles. In NDN, communication is driven by content names rather than endpoint addresses. A sample *interest* and corresponding *data* packet with header fields relevant to our analysis is shown in Figure 1. To retrieve content (e.g., to visit a website), a user acting as a data consumer sends interest packets that contain a hierarchical *name* which identifies the desired data.

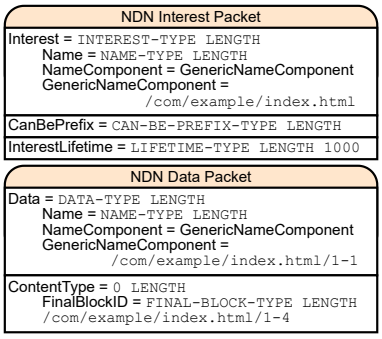

Figure 1: An NDN interest packet from a user requesting index.html and the associated segmented data packet.

When an intermediate NDN router receives an interest, it stores the interest's content name within a *pending interest table*. The purpose of this table is to associate content names with the interfaces from where interest(s) for that content arrived. Next, the NDN router performs longest-prefix matching using the *name* listed in the interest and its *forwarding information base* (similar to a routing table) to determine the correct outgoing interface.

Eventually, the data packet(s) that match the interest's content name are found. The data can be obtained from its original host (i.e., the owner of `example.com`) or found in the *content store* (i.e., cache) of any NDN router that recently served the data and chose to save a copy locally. The data packet(s) are then returned to the original host that issued the interest packet along the reverse path established by entries in each router's pending interest table.

NDN simulation testbeds. NDN6 [45] provides a globally distributed NDN hardware testbed but it primarily serves to experiment with network connectivity and lacks the flexibility required for large-scale controlled experimentation. In contrast, pairing the NDN-CXX [65] and MiniNDN [75] libraries provides support for local experimentation with fine-grained control over network topologies, and is supported by the evolving NDN development ecosystem [66]. While NDN-CXX provides data-plane and control-plane functionality, MiniNDN uses these capabilities for building an interconnected network of devices sitting atop of Mininet [56].

Visiting NDN websites. Earlier research efforts have explored the implementation of web browsers tailored for NDN [68, 76, 83, 100]. However, these prototypes are no longer maintained and have been folded into projects such as NDNts [46] and NDNTools [67]. While NDNts is tightly coupled with the NDN6 testbed (and hence, sub-optimal for the purposes of our study), NDNTools provides a flexible set of high-level primitives that can be used to serve/retrieve files over MiniNDN/NDN-CXX. Unfortunately, NDNTools does not natively provide application-layer functionality resembling modern web browsing. To address this gap, we integrate NDN functionality within a modern web browser (§3.2).

Research Question 1. How can modern browser-driven website accesses be reproduced over NDN for WF evaluation?

Table 1: Recent quantitative NDN security evaluations.

Paper	Year	Focus	Network	Runtime	Dataset
[64]	2019	Traffic Analysis	5 Nodes	10 seconds	6 traces
[9]	2021	Traffic Analysis	7 Nodes	600 seconds	19,626 traces
[113]	2021	Cache Pollution	7 Nodes	60 seconds	3 traces
[36]	2024	Cache Pollution	41 Nodes	106 seconds	10 traces
[47]	2024	Cache Pollution	9 Nodes	2.5 hours	9 traces
[60]	2024	Interest Flooding	24 Nodes	300 seconds	unknown
[53]	2025	Interest Flooding	11 Nodes	300 seconds	200 traces
Our	–	Encrypted Traffic	**257 Nodes**	**500+ hours**	**80,000 traces**

2.2 Existing NDN Security & Privacy Analyses

Existing research addressed various aspects of NDN security and privacy, but has largely overlooked how adversaries may monitor and/or disrupt users' access to data [103].

Limited scope on traffic analysis. Broader surveys [27, 39, 80, 81] and experimental works addressing NDN security threats (see Table 1) mostly focus on infrastructure-level attacks (e.g., cache pollution [113], interest flooding [53]). Relevant exceptions include the works of Bardhi et al. [9] and Mozaffari et al. [64], which examine traffic analysis threats in NDN. These works assume that network participants (i.e., data consumers and producers) do not use privacy-enhancing technologies, and consider adversaries capable of inspecting the *Name* field in unencrypted interest packets and/or a data producers' public keys and signatures tied to data packets. These works shed light on how adversaries can profile user activity and/or enforce content blocking.

Narrow evaluation scale. Most quantitative security evaluations on NDN encompass small-scale NDN deployments in tight time-restricted experiments, as highlighted in Table 1. These constrained runtime setups limit their applicability for studying long-term adversarial strategies. These aspects are especially important in encrypted traffic analysis attacks like WF, where persistent observation enables adversaries to exploit long-term patterns for accurate inference. Our work addresses this gap by introducing a scalable simulation framework (§3) designed to support large-scale, long-running encrypted traffic analysis experiments across arbitrary NDN topology configurations (§3.1) with realistic web browsing and background traffic (§3.5). Although no formal network size, trace count, or runtime threshold defines "large-scale", we use the term comparatively with the existing evaluations listed in Table 1.

Research Question 2. Can NDN experiments be scaled in topology, duration, and volume to support modern WF evaluation?

2.3 Privacy-Enhancing Technologies in NDN

In the TCP/IP ecosystem, users often rely on PETs such as Tor [23] to anonymize their communications. Tor implements metadata obfuscation mechanisms [90] that aim to mitigate traffic analysis.

Anonymity systems on NDN. Analogous systems have been developed to address privacy concerns in the content-centric paradigm of NDN. ANDaNA [22], AC3N [95], and PRIDN [105] use an onion routing paradigm reminiscent of Tor, redesigned to be entirely functional using NDN network primitives. These tools are designed to protect the identity of data consumers while concealing the contents they are interested in accessing without requiring any cooperation from the producers of the data.

More advanced solutions aim to provide anonymity for both data consumers and producers. Al Azad et al. [7], Kita et al. [49], and Aldaoud et al. [2] explore mutual anonymity schemes, often assuming cooperative environments in which content producers actively participate in the protocol. While these solutions offer strong privacy guarantees, they rest on cooperation assumptions that may be challenging to secure in real-world scenarios; for instance, censorship-resistant access to Tor via domain fronting has been disabled in the past by large CDN providers [101].

ANDaNA. Among the above NDN-focused anonymity systems, ANDaNA [22] stands out as the most influential effort to replicate Tor-like behavior in a content-centric setting. Like Tor, ANDaNA forwards requests through multiple relays using layered encryption: the entry cannot learn the content name, while the exit cannot learn the requester. Unlike Tor's circuit-based streams, every hop is an NDN interest–data exchange. We further detail our implementation of ANDaNA and its inter-relay data exchange mechanisms in §3.3. Given ANDaNA's conceptual closeness to Tor, we focus on it as the primary NDN-deployed PET in our study. This allows us to evaluate whether WF defenses shown to be effective on Tor can achieve similar performance in an NDN-based setting (§3.6).

NDN architectural implications on traffic analysis. NDN introduces architectural features that may introduce new observable phenomena such as cache hits [48] or lookup failures during content retrieval [10] that could limit the success of traffic analysis attacks. Additionally, NDN's symmetric request-response model and segmentation of content into named chunks can cause traffic fragmentation and interleaving, analogous to Tor's stream multiplexing [103]. While these features suggest potential complications for adversaries, no prior work has assessed whether NDN's design can limit the success of traffic analysis attacks.

Research Question 3. To what extent do NDN's architecture and ANDaNA reduce susceptibility to WF?

2.4 Website Fingerprinting

WF is an encrypted traffic analysis attack that seeks to identify which websites a user visits by analyzing network metadata.

Threat model. A typical WF threat model has the adversary positioned as a passive observer who monitors the encrypted network flows exchanged by the target user's device; these flows traverse some encrypted tunnel that shrouds the user's Internet destination. The fundamental assumption behind WF attacks is that each website, due to its unique layout, resource loading behavior, or server configuration produces a distinct traffic metadata signature.

WF attacks are typically evaluated under two settings: *closed-world*, where the user is assumed to access only a fixed set of monitored websites, and *open-world*, where the user may visit any available (and unmonitored) website and the adversary aims to identify accesses to the set of monitored websites. Additionally, more complex multi-tab browsing [20, 43, 62] settings are common in which multiple webpages are fetched concurrently, causing their network traces to interleave. In this paper, we evaluate NDN fingerprinting in the closed and open-world settings on single and multi-tab data.

WF attacks. Recent advancements in WF have been driven by DL techniques which can learn complex hierarchical and temporal features from raw traffic [8]. Prominent attacks for single-tab WF settings on both closed and open-world datasets include Deep Fingerprinting [88], Tik-Tok [78], and Robust Fingerprinting [84]. Expanding our scope to the *multi-tab* setting, DL models like BAPM [32], ARES [20], TMWF [43], and CountMamba [21] innovate on both data representation and neural network architecture.

Other sophisticated attacks, also relying on DL, operate in more complex scenarios involving training data scarcity [72, 89, 107], subpage identification [112], website identification via visits to subpage sets [63], or website identification on early page loading [19, 21].

WF defenses. Many defenses [61, 106] have been proposed to mitigate WF attacks by distorting the traffic patterns and observable features (e.g., packet sizes, times) that classifiers depend on.

Padding strategies are among the most widely studied class of defenses. CS-BuFLO [12] and Tamaraw [13] enforce fixed-size packet transmissions at uniform intervals. WTF-PAD [44] and FRONT [29], inject dummy packets in an adaptive or randomized fashion. Timing-based padding defenses, such as RegulaTor [38], focus on shaping packet bursts and inter-packet delays. Alternative designs such as traffic-splitting defenses [35, 55], trace-merging defenses [29], adversarial learning [42, 69, 77, 82] and generative approaches [30] have also been shown to be effective.

In this work, we focus on padding strategies since these remain core benchmarks for recent WF attacks [19, 21, 84]. In §4, we discuss how re-purposing these existing WF defenses (designed for Tor over TCP/IP) to the NDN context is non-trivial, as they all violate NDN's core principles around stateful forwarding and data retrieval.

WF across networking contexts. While WF has been studied in other emerging network environments such as LTE [51], 5G [109], and LEO satellite communication systems [87], these efforts have focused primarily on the variability in link-layer characteristics of TCP/IP. In contrast, our work builds up to investigate WF in a fundamentally different architectural setting, specifically NDN (§3).

Research Question 4. To what extent do WF attacks and defenses transfer to NDN, and are any adaptations required?

3 Website Fingerprinting Testbed for NDN

NDN currently lacks large-scale experimental platforms capable of simulating realistic web access patterns, particularly under conditions required for studying encrypted traffic analysis. To bridge this gap, we develop a custom testbed that emulates modern application-layer behavior over NDN and enables the systematic evaluation of WF attacks in this setting. Yet, we do not assume that NDN will replace TCP/IP wholesale or that future NDN-native applications will reproduce the request behavior of today's browsers. NDN can instead be introduced incrementally, including within targeted deployments or alongside and over IP [111]. We therefore examine one concrete use case: browser-driven, web-like named-content retrieval over a controlled NDN deployment. This workload allows us to study whether established WF methods transfer to named-content delivery, but our conclusions do not assume that every future NDN application will generate the same traffic patterns.

Our testbed, outlined in Figure 2, is the first to support WF experiments over NDN, moving beyond coarse-grained traffic analysis endeavors (§2.2). Its modular structure allows the components responsible for network topology generation, PET deployment, and other tasks to operate independently. These components can be modified or extended, e.g., to enable other traffic analysis tasks (§2.2) or to study additional real-world behaviors. Furthermore, (but not evaluated in this work) our testbed supports flexible assignment of node locations to exercise geographical diversity (§3.1) and replicating real-world website content as NDN-named resources over long-running tests (§3.4). These features can also support future work on the analysis of distributional shifts in network traffic (e.g., geographical, temporal).

Code/data availability. We release our source code and datasets as open-source to support reproducibility and future research [102].

3.1 Network Topology Generation

To address the topology component of RQ2 (§2.2), we developed a representative network topology that reflects the connectivity, data flow, and caching dynamics inherent to NDN.

Creating a network topology. We leverage the RocketFuel [91] dataset containing reconstructed router-level topologies of real ISPs. Among the available topologies, our evaluation focuses on the *Ebone* ISP (AS 1755). The resulting graph preserves the structure of the ISP network, although it should be understood as a historical, measurement-derived topology covering Western Europe with (i) router-level granularity, (ii) link-delay data, and (iii) human-readable interface names. The RocketFuel internet service provider topology dataset has been used by other recent NDN studies [14, 54] and TCP/IP simulations [41, 71].

Adding metadata to links. We also attach metadata to every link (namely latency and routing weight). When the two router IDs of a particular link can be resolved unambiguously, we label it directly using the RocketFuel data. When the two router IDs are incomplete or inconsistent, we fall back to location-based approximation. The result is a connected, weighted, undirected graph that captures both structural and geographical aspects of the measured network.

Assigning roles to interfaces. To support our web browsing use-case for the topology, we rename node identifiers according to our required quotas. One Domain Name System (DNS) server (`dns`), one target user (`tu`), and a fixed number of background users (`u*`), anonymization relays (`r*`), and website servers (`s*`), with the remaining vertices assigned to a neutral host bucket (`h*`). Each simulated node runs the NDN Forwarding Daemon and can participate in forwarding and caching, while the content and NDN-DNS servers additionally advertise their assigned name prefixes.

Instead of randomly assigning roles to nodes, we infer coarse attachment roles using hints from human-readable interface names based on conservative and dataset-aware patterns. Interfaces with labels that indicate name service (e.g., `dns\d*`, `ns\d+`) yield DNS assignments. Interfaces names associated with service infrastructure (e.g., `host\d*`, `www\d*`, `web\d*`, etc.) yield server assignments. Customer-facing or access-oriented markers (e.g., `cust\d*`) yield user assignments. This results in a topology (fully illustrated in Appendix A) that is faithful to RocketFuel's measurements and that is annotated to support simulation.

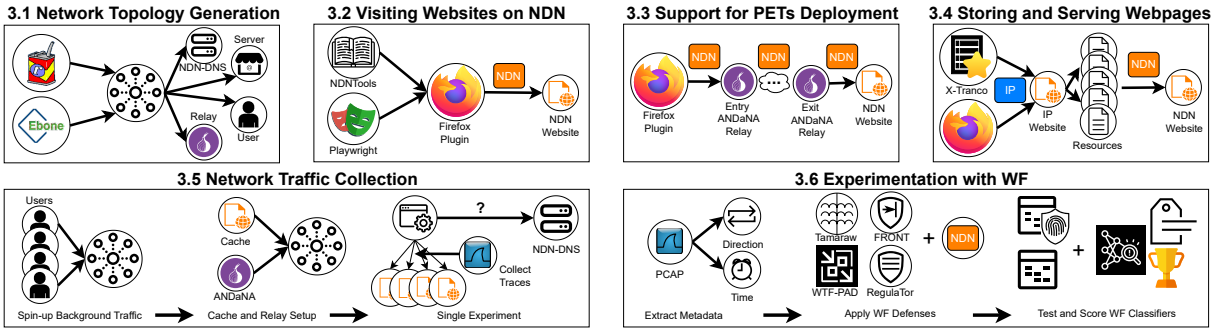

Figure 2: Overview of the components and operational workflow of our NDN website fingerprinting testbed

Topology boundary. All users, relays, content servers, and advertised prefixes reside within a single simulated ISP (*Ebone*). Hence, the testbed captures variation arising from intra-domain paths, link delays, competing traffic, and cache state, but it does not reproduce global origin geography, CDN placement, interdomain routing, or worldwide latency variation. Our results should consequently be interpreted as measurements from an independent ISP-scale NDN deployment, rather than a world-scale NDN Internet.

3.2 Visiting Websites on NDN

To address RQ1 (§2.1), we developed a plugin for *Firefox* that allows it to directly request and load webpages over NDN by using the *Playwright* [28] automation framework. Firefox functions as a standard web client at the user interface level, while the underlying transport and delivery mechanisms are performed on NDN.

Accessing web content over NDN. At runtime, we capture HTTPS requests issued by the browser and repackage them into NDN *interest* packets. Each intercepted request is mapped to a unique NDN name derived from its corresponding URL, and is then forwarded to the NDN network. In cases where the authoritative NDN namespace of the URL is unknown, our plugin automatically issues an NDN-DNS [1] request to resolve the appropriate prefix.

The corresponding NDN *data* packets are reassembled by the plugin to form a complete HTTPS response. The assembled response is then passed back to the browser, enabling the webpage to render normally. In this manner, the plugin effectively translates between the request–response abstraction of HTTPS and the name–data exchange model of NDN, providing a transparent bridge between web application logic and NDN transport semantics.

Loading a webpage. In our evaluation, webpages are accessed dynamically with Playwright's automated navigation capabilities to enable realistic testing conditions in which browser-initiated HTTPS requests are issued naturally during rendering.

3.3 Support for PETs Deployment

A WF evaluation is meaningful when the target can conceal direct identifiers of the websites they visit. Without a PET, an on-path adversary could simply inspect the content names carried by outgoing NDN interests and trivially observe any visited website. We therefore evaluate WF after ANDaNA [22] is deployed to hide the requested names and data. To visit a webpage over ANDaNA the user must perform the following steps:

1. Circuit initialization. A user first selects a sequence of n relays from a public list of relays that includes each relay's public key. The user then generates distinct AES-256 symmetric keys, K_i for each relay, R_i, in the circuit and distributes them securely by encrypting each symmetric key with the corresponding relay's public key to form a secure ephemeral session.

2. Onionized content requests The target user constructs an interest packet embedding a unique session identifier (*sid*) within the content name (we use unix timestamps for *sid*, as no formal *sid* selection process is provided by ANDaNA's specification). The user encrypts their interest as $\mathsf{Enc}_{K_1}(\ldots\mathsf{Enc}_{K_n}(\text{Interest}))$. Then, the encrypted interest is routed through the relay sequence, with each relay "peeling" back one layer of encryption. At the exit relay, the interest is released to the broader NDN network as a standard interest packet. The data packets corresponding to each of these three pending interests are incrementally satisfied and encrypted in reverse order: consumer–entry, entry–exit, and exit–content.

Tor ordinarily builds a three-relay circuit over persistent TCP/TLS connections. ANDaNA does not establish a connection between the consumer and exit. Instead, its "circuit" is a per-interest ephemeral relay, and each onion-routing step is itself a separately routed NDN interest–data exchange. ANDaNA's security analysis notes that due to the absence of source and destination addresses in NDN packets, two relays are sufficient to provide consumer–producer unlinkability, offering equivalent protection to three-relay Tor paths [22].

Reimplementation of ANDaNA. Because the original ANDaNA implementation is no longer maintained, we implemented its operation using current MiniNDN, NDNTools, and NDN-CXX tooling. We pad packets so that length is uninformative, and consequently retain only timing and direction in the WF representation [99]. The original ANDaNA description treated interest and data as packet types, but modern NDN forwarders can return *NACK*s if upstream nodes cannot satisfy an interest. Our implementation uses NACKs to denote a failed hop-level retrieval, and allows the affected segment to be re-transmitted.

3.4 Storing and Serving NDN Webpages

To support our analysis, we replicate the file structure of popular real-world websites as NDN named contents, and simulate their retrieval over our NDN testbed.

Selecting websites. We use X-Tranco to obtain a stable ranking of popular domains [74], noting that this does not imply that today's most popular websites would be deployed unchanged over future NDN deployments. However, these websites represent reproducible, complex workloads containing the media and dependent requests that produce website-specific traffic patterns, allowing our dataset to remain comparable with conventional WF evaluations.

Profiling websites' contents. We used Firefox to visit the top domains from the X-Tranco list [74] as of September 2025. We recorded this process in a `har` file, which contains the sequence of HTTPS requests/responses issued by the browser to load the page. We excluded pages from the X-Tranco list that were inaccessible (e.g. content delivery networks or DNS servers) or those that did not trigger the Firefox `load` API event within 30 seconds. We used the top-100 websites as our set of monitored pages (i.e., the *closed-world* dataset), and expanded to the top-10,000 as our set of unmonitored pages (i.e., the *open-world* dataset).

Serving webpages. To make conventional TCP/IP web content available within our NDN architecture, we created a preprocessing pipeline that converts `har` recordings into hierarchically named NDN data objects. For each target domain, the associated `har` archive is parsed to extract an ordered sequence of HTTPS responses. Each response is stored independently and assigned a unique NDN name derived from the originating URL. This process effectively reconstructs the file structure of a conventional website as a collection of addressable NDN contents, preserving both object granularity and retrieval ordering. During runtime, our Firefox plugin compiles and sends NDN interests for the URLs on HTTPS requests issued by the browser in real-time, allowing the user to automatically retrieve webpage components over NDN.

Parallel update and retrieval. Modern webpages (especially those in the top-lists) often employ asynchronous and time dependent HTTPS requests like session tokens, embedded advertisements, real-time scripts, etc. which complicate deterministic replay and are difficult to reproduce faithfully from archived data alone. Furthermore, when replaying these kinds of HTTPS requests from `har` archives, Firefox frequently fails to load a webpage due to expired cookies, embedded nonce-bearing URLs, or dynamic query parameters used by CDNs for cache-busting, etc.

To preserve realistic loading behavior while maintaining NDN semantics, we adopt a hybrid retrieval strategy. In parallel to the NDN interest-data retrieval, our plugin can be configured to issue a live HTTPS request over TCP/IP to refresh a particular resource that has become invalid. The up-to-date content then replaces the cached NDN data as the page is loaded. Beyond improving reliability, this mechanism also allows our dataset to naturally capture potential changes that occur in real websites over time.

3.5 Network Traffic Collection

To address the network traffic collection component of RQ2 (§2.2), we developed a scalable and automated three-stage website fetch process to produce `pcap` traces reflecting the NDN traffic observable by an adversary.

Network assembly. During this initialization phase, MiniNDN instantiates the generated network topology. Each host operates as a lightweight Linux process running the NDN Forwarding Daemon, with routing tables configured using a link-state protocol based on Dijkstra's algorithm with metadata (e.g., delay, routing weight) derived from the RocketFuel dataset (§3.1). Server nodes advertise NDN data names corresponding to hosted website content, enabling *interest* packets from any node in the network to be efficiently routed to the appropriate content provider.

Background traffic spin-up. Before each webpage fetch, the testbed performs a warm-up phase to introduce variability in the network cache state, background traffic intensity, network stability, and ANDaNA routing. During this period, all non-target user nodes periodically issue requests for random webpage content (sampled from either the open- or closed-world sets) with concurrently and independently sampled delays and resource sizes. This background activity creates a dynamic network environment in which packets can experience arbitrary queuing delays, retransmissions, or losses due to congestion or competing cache access. Thus, each website access trace contains the dynamic effects of congestion and timing variance that would naturally arise in a live NDN deployment.

Request a website. First, if ANDaNA is enabled, two relays are selected without replacement from the pre-determined pool of participating nodes (§3.3) to serve as entry and exit points. Then, the target user accesses a particular webpage in Firefox using our plugin (with or without ANDaNA). All resulting packet exchanges are captured into `.pcap` files using `tcpdump`, yielding a trace representative of the traffic visible to an on-path adversary.

3.6 Experimentation with WF

We now describe how raw website traces are preprocessed into formats compatible with existing WF tooling, and how we apply WF attacks and defenses to our data.

Extracting metadata from webpage fetches. Given a `.pcap` recording obtained from a webpage access, we first extract metadata from the trace, namely packet direction and time. The result is a sequence of floating point values representing packets observed on the wire where magnitude encodes inter-arrival time and sign denotes direction (e.g., `[0.1,-0.2,···]`). This representation retains the temporal and directional characteristics of a website trace while omitting payload content which remains encrypted, and packet lengths which are uninformative due to padding.

Creating multi-tab traces. To generate traces representative of a multi-tab browsing scenario, we adopt Jin et al.'s [43] methodology of merging independently recorded single webpage visits into composite sessions. For each sample, a random subset of traces is trimmed so that the total length does not exceed 10,000 packets, then recursively merged with overlap ratios uniformly sampled between 0% and 40%. A *purity threshold* of 20% ensures that at least one-fifth of each trace remains non-overlapping.

Simulating defended traffic. To assess the robustness of WF attacks against countermeasures, our testbed supports the integration of WF defenses (§2.4) on top of ANDaNA through defense simulators [31]. These simulators, typically provided by existing defenses' designers, apply defense-specific transformations to previously collected undefended traces, producing traces that reflect the theoretical behavior of each defense technique. Prior work [31] showed that such simulations accurately reflect these defenses' real-world implementation performance.

4 WF Defenses in the NDN Context

To address the defense-transfer and adaptation component of RQ4, we show that prominent padding-based WF defenses designed for TCP/IP cannot be applied directly to NDN's content-centric model and propose modifications that make them compatible with NDN. We further validate these claims during our evaluation in §6.

4.1 NDN Communication Semantics Violations

As described in §2.1, NDN adheres to a strict interest/data exchange model: every interest is expected to return data packet(s) within a bounded timeout window and each data packet must be preceded by a corresponding interest. Unlike TCP/IP, where bidirectional traffic flows can be decoupled and arbitrarily padded, NDN enforces symmetry at the network layer. Applying padding-based defenses (e.g., Tamaraw [13], FRONT [29], WTF-PAD [44], RegulaTor [38]) without accounting for these constraints leads to three classes of NDN protocol violations (illustrative examples in Appendix B).

(1) Unsolicited data packets. WF defenses in NDN cannot arbitrarily create dummy data packets. NDN routers forward data packets only if a matching interest exists in their *pending interest table*. If a router receives a data packet with no corresponding interest, the data is discarded. All the defenses we examine attempt to add dummy packets into the incoming direction (i.e., data) regardless of whether a dummy outgoing packet (i.e., interest) was added beforehand. The assumption that servers can arbitrarily emit any amount of traffic towards a client (given a pre-established connection) is valid in TCP/IP but breaks down in NDN.

(2) Unsatisfied interest packets. WF defenses in NDN cannot add dummy interest packets without reflecting a corresponding data reply. In NDN, when a router cannot satisfy a particular interest, (i.e., if no data is returned after forwarding the interest) the router may issue a *NACK* (Negative Acknowledgment) packet. These returned packets can be observed and identified by a WF-capable adversary, who could then weed them (and the corresponding dummy interest) out during the analysis of NDN traces.

(3) Interest timeout sensitivity. WF defenses in NDN cannot arbitrarily delay data packet arrival times. In TCP/IP, such delays are typically inconsequential beyond latency/throughput penalties. However, in NDN, interests are only retained by each router's *pending interest table* for a pre-determined time-to-live, defaulting to 4 seconds when omitted [70]. Longer lifetimes can be configured by network administrators and/or users, and we evaluate up to extreme values of 30 seconds. WF defenses that delay data packet(s) corresponding to a particular interest beyond the interest's TTL will cause the associated data to be lost.

4.2 Content-Centric WF Defense Strategies

We now propose a set of *content-centric* (CC) variants for four prominent padding-based WF defenses: CC-Tamaraw, CC-FRONT, CC-WTF-PAD, and CC-RegulaTor. We keep our CC-variants as faithful to the original defenses as possible, guided by three objectives: *(i)* Retain the primary obfuscation mechanism and design goals of the original defense; *(ii)* Ensure that the three core protocol violations introduced in §4.1 are not infringed by the defense (we validate these changes are necessary in §6.4); *(iii)* Modify each defenses' scheduling algorithm to better obfuscate timing features, which we experimentally observed in §6.2 to be more used by WF classifiers in NDN compared with direction (valuable for TCP/IP). We validate these performance improvements in §6.5 and §6.7. Our CC-variants are representative compatibility-preserving designs, rather than the only conceivable set of NDN defenses (see §7).

CC-Tamaraw. Tamaraw regularizes traffic by forcing outgoing and incoming packets to follow pre-determined schedules, and sending dummy packets at the scheduled times when no real data is needed. We adapted Tamaraw to create stricter constraints on its configurable parameters, ensuring that CC-Tamaraw will always generate a packet schedule that represents valid NDN traffic.

CC-Tamaraw's scheduler defines parameters for an interest interval time, i, and data interval time, d. We enforce three constraints on the values of i and d as follows: First, to prevent unsolicited data before the first scheduled interest (at time i), we suppress data packets scheduled at times:

$$t \in \left\{ d \cdot c \;\middle|\; c \in \mathbb{Z},\ 1 \le c < \left\lfloor \frac{i}{d} \right\rfloor \right\}.$$

Second, to guarantee the unsatisfied interest requirement, i and d must satisfy $i > d$ and $\frac{i}{d} \notin \mathbb{Z}$. These constraints are required because if $d > i$, more interests would be scheduled than there are corresponding data packets. $i = kd, k \in \mathbb{Z} > 0$ represents an edge-case where, depending on the exact number of packets in a trace, there could be up to one interest packet without any data. Third and lastly, to ensure that no interests timeout, we require that $\frac{i}{d} < interest\_timeout$.

CC-FRONT. FRONT independently samples a number of incoming and outgoing dummy packets (i.e. the *budget*) and concentrates them near the beginning of the trace. Independent sampling is valid over a TCP connection, but when applied to NDN, FRONT can violate the unsatisfied interest and unsolicited data requirements. CC-FRONT instead samples the dummy interest budget first, then samples the dummy data budget *conditionally* as ≥ to the the interest budget. Furthermore, to better hide timing information CC-FRONT introduces a delay layer based on the Rayleigh distribution:

$$f(x;\sigma) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right), \quad x \ge 0,$$

with timestamps sampled from:

$$x = \sigma\sqrt{-2\ln U}, \quad U \sim \mathrm{Unif}(0,1).$$

Each dummy data packet is delayed relative to its corresponding interest. This preserves NDN semantics while obfuscating timing features (e.g., packet concentration bursts or stable packet-per-second timing intervals) in the real data packets associated with a particular interest. Choosing the Rayleigh distribution was a design choice rather than an NDN requirement. Uniform, exponential, or other distributions could also produce valid NDN exchanges. We use Rayleigh because its early timestamp concentration and trace-to-trace randomness are the defining intuition behind FRONT.

CC-WTF-PAD. WTF-PAD identifies bursts and gaps using a finite state machine and injects dummy packets during gaps to obscure distinctive traffic timing. Directly applying WTF-PAD to NDN causes unsolicited data and unsatisfied interests. CC-WTF-PAD preserves the state machine's decision about *when* dummy padding events should occur, but ensures that all dummy padding is valid

NDN traffic. At each padding event, CC-WTF-PAD creates dummy interests first, then samples a number of dummy data packets following a triangular distribution:

$$f(x; a, b, m) = \begin{cases} \frac{2(x-a+1)}{(b-a+1)(m-a+1)}, & a \le x \le m \\ \frac{2(b-x+1)}{(b-a+1)(b-m+1)}, & m < x \le b. \end{cases}$$

Here, a, b, and m define the minimum, maximum and mode number of dummy data packets, respectively. Like CC-FRONT, the exact distribution was a design choice rather than a strict protocol requirement. We chose triangular because it creates dummy packets that obscure timing structures valuable to WF classifiers like packet inter-arrival times and burst lengths. Furthermore, its parameters allow for tuning overheads, while also introducing randomness to avoid creating a consistently observable pattern.

CC-RegulaTor. Unlike FRONT and WTF-PAD that rely on padding alone, RegulaTor introduces temporal regularization by adjusting inter-packet gaps. Therefore, RegulaTor can violate all three categories of NDN protocol requirements. Similarly to CC-FRONT, we ensure that the randomly sampled dummy data budget is *dependent* on and ≥ than the interest budget. Unlike CC-FRONT, RegulaTor treats these budgets as upper bounds rather than fixed counts, so CC-RegulaTor injects additional dummy data for each otherwise unmatched dummy interest. Like CC-WTF-PAD, additional dummy data packets are folded into sequences determined by RegulaTor's algorithm. To prevent timeout violations, CC-RegulaTor also bounds the intra-packet gap used for temporal smoothing:

$$gap < \frac{interest\_timeout}{queue\_size + (budget - already\_padded) + 1}$$

Restricting the intra-packet gaps admits a design trade-off. Users can increase the maximum from NDN's 4 s default, so we evaluate settings up to 30 s rather than treating one timeout as architecturally fixed. We will show that temporal smoothing is far more effective in stopping WF attacks in NDN compared to CC-FRONT and CC-WTF-PAD (see §6.5).

Rationale for required and flexible adaptations. The CC-design adaptations of pairing dummy interests and data, as well as completing each exchange before timeouts, are required (i.e., "forced") by NDN. In contrast, retaining FRONT's Rayleigh window, using bounded triangular sampling at state-machine-determined intervals in CC-WTF-PAD, and selecting the smoothing bound in CC-RegulaTor were all flexible choices deliberately made to preserve each defense's original intent.

5 Experimental Testbed and Data Collection

To evaluate the effectiveness of WF attacks and defenses in NDN under varying network conditions, we evaluate a matrix of 4 datasets, 7 WF attacks, and 4 WF defenses, detailed in this section.

5.1 Testbed and Dataset

Network topology. We used the RocketFuel topology with the exact same 257 nodes, 41 users, 21 servers, and 10 relays for every experiment. As discussed in §3.1, one user acts as the *target* user, and one server hosts the NDN-DNS service. The remaining 185 nodes serve as neutral forwarding routers. Each website was hosted at one content server, and websites were assigned to servers in their X-Tranco selection order using round-robin placement. This placement models a controlled single-ISP NDN deployment rather than the websites' actual geographic origin or CDN deployment.

Network conditions and deployment. We configure the network with two traffic-stability levels: *sparse* and *dense*. Both configurations included 40 users generating background traffic, with file sizes sampled from monitored or unmonitored resources between 100 KB and 20 MB. In the sparse setting, users waited 3000–9000ms between requests while in the dense setting, they waited 500–1500ms. We experimentally observed that the *target* user experienced packet loss between 1.5–2.5% with an average RTT of ~27 ms in the sparse setting which increased to 2–3.5% with an average RTT of ~34 ms in the dense setting.

Although each website was stored entirely on one server, content from background users' interests were being cached at the 185 neutral forwarding routers. Thus, during data collection, background users simultaneously randomized network congestion artifacts and content retrieval times based on if/where content was cached. To promote greater fluctuations in these artifacts, we configured the background users with a 50% increased probability of selecting data from the website currently being visited by the target. We collected four datasets in total: two under NDN and two under ANDaNA. In each ANDaNA trial, the target user randomly selected 2 relays from the pool of 10.

Dataset collection. We collected 80,000 single-page loads covering 10,100 unique webpages across four network configurations: sparse and dense background traffic in NDN and ANDaNA. We used the same disjoint sets of 100 monitored and 10,000 unmonitored webpages in every configuration. Within each configuration, the target user visited each monitored webpage 100 times, producing 10,000 monitored traces, and each unmonitored webpage once, producing another 10,000 traces. Each configuration therefore contains 20,000 independently collected `pcaps`, and the complete collection contains $4 \times (100 \times 100 + 10,000) = 80,000$ traces.

Following Jin et al. [43], we construct each multi-tab evaluation dataset by merging these independently collected traces into 50,000 synthetic three-tab traces, as described in §3.6.

Laboratory testbed hardware. All datasets were collected on Google Cloud `c4a-standard-16` VMs (16 vCPUs, 64 GB RAM). Completing the 80,000 page loads required approximately 500 hours when summing the collection runtime across all four configurations. This figure represents live collection time, not the elapsed wall-clock time or the total time recorded in `pcap` files.

5.2 WF Attacks and Defenses

WF libraries. In our experiments, we evaluated three single-tab WF attacks: Deep Fingerprinting (DF) [88], Tik-Tok (TT)[78], and Robust Fingerprinting (RF) [85], three multi-tab WF attacks: BAPM [32], TMWF [43], and ARES [19], and one attack for both single and multi-tab data: CountMamba (CM) [21]. All the WF attacks and padding-based defenses (Tamaraw, FRONT, WTF-PAD, RegulaTor) we considered had open-source Python implementations available. Unless noted otherwise, we use the recommended set of parameters for attacks and defenses from their authors.

WF attacks and implicit CC-awareness. We do not apply "frozen" classifiers trained on TCP/IP traffic directly to NDN. For each NDN/ANDaNA and defense configuration, we train a separate classifier. The attacks are therefore *implicitly CC-aware*: they can learn the observable consequences of CC operations whenever those operations alter the traffic metadata. Interest–Data exchanges, NACKs, retransmissions, content segmentation, and cache-dependent retrieval can affect packet direction, inter-arrival times, packet counts, and burst structure, even though the classifier is not told which protocol event produced a particular pattern. The environment-specific features and learned representations observed in §6.2 are consistent with this form of implicit adaptation.

WF defenses and features. Our defense evaluation is limited to padding mechanisms that disrupt packet direction, timing, and burst patterns rather than NDN-specific protocol traits. Prior NDN privacy tools explored how to conceal such traits from an on-path adversary; for instance, Harpocrates [7] and PRIDN [105] obscure interest/data exchange behavior, NACKs, or retransmissions by spreading content requests across a distributed network of routers.

6 Experimental Results

In this section, we first demonstrate that existing TCP/IP-based WF attacks remain effective over NDN (§6.1), despite using fundamentally different internal representations of website traces (§6.2/§6.3).

Evaluation metrics. For each scenario, we defined a multi-class classification task in which the goal was to identify the correct website label (among 100 monitored sites) or determine that the trace belonged to the unmonitored open-world set. Each dataset was divided into an 80/10/10 training/validation/test split. In the single-tab setting, the classifier outputs one label, and we report overall accuracy on the test set. In the multi-tab setting, the classifier outputs n predicted labels for n concurrently loaded sites, and we evaluate Precision@n (P@n), that is, the proportion of top-n predictions that correspond to the true sites present in the trace. These validation and evaluation metrics are standard in many existing WF evaluations [19–21, 43]. The reported accuracies include 95% binomial proportion confidence intervals (CIs).

6.1 NDN's Architecture Does Not Prevent WF

To address RQ3 (§2.3) and the attack-transfer component of RQ4 (§2.4), we determine whether existing WF attacks can succeed on traces collected over NDN when no WF-specific defenses are deployed. We assume that the target user either: a) partially encrypts the content name they requests and that the content payload is encrypted (e.g., with [15, 34, 37], akin to TLS) or; b) leverages an onion routing-based privacy-enhancing technology (*ANDaNA*). We evaluate the WF attacks by providing them with sequences containing the direction (and timing, when compatible) of all the packets entering and leaving the target user's machine. The results of this experiment are shown in Table 2.

Single-tab WF attacks succeed on NDN. In the single-tab closed-world dataset with *sparse* background traffic, a WF adversary can achieve at least 98.0% accuracy (using TT). With the additional protection of ANDaNA relays, the adversary still attains a minimum accuracy of 98.0% (using DF). In the open-world setting, accuracy remained extremely high, 98.6% for NDN and 98.0% for ANDaNA (both using DF). CI overlaps across different classifiers indicate that their performance differences are marginal. We attribute this observation to the proximity of the classifiers to perfect accuracy and the ease of operating on undefended data.

Table 2: WF attack performance and 95% binomial CI for all four datasets without defenses.

Attack	Sparse Background		Dense Background	
	NDN	ANDaNA	NDN	ANDaNA
Closed-World Only, 1 Tab Accuracy				
DF	$98.5\%^{\pm 0.75}$	$98.0\%^{\pm 0.87}$	$98.2\%^{\pm 0.82}$	$97.9\%^{\pm 0.89}$
TT	$98.0\%^{\pm 0.87}$	$98.1\%^{\pm 0.85}$	$98.4\%^{\pm 0.78}$	$98.0\%^{\pm 0.87}$
RF	$98.9\%^{\pm 0.65}$	$98.3\%^{\pm 0.80}$	$98.2\%^{\pm 0.82}$	$98.1\%^{\pm 0.85}$
CM	$99.8\%^{\pm 0.20}$	$98.4\%^{\pm 0.78}$	$98.7\%^{\pm 0.70}$	$98.3\%^{\pm 0.80}$
Open-World, 1 Tab Accuracy				
DF	$98.6\%^{\pm 0.51}$	$98.0\%^{\pm 0.61}$	$98.0\%^{\pm 0.61}$	$97.9\%^{\pm 0.63}$
TT	$98.7\%^{\pm 0.50}$	$98.1\%^{\pm 0.60}$	$98.0\%^{\pm 0.61}$	$97.9\%^{\pm 0.63}$
RF	$98.8\%^{\pm 0.48}$	$98.2\%^{\pm 0.58}$	$98.5\%^{\pm 0.53}$	$98.0\%^{\pm 0.61}$
CM	$99.0\%^{\pm 0.44}$	$98.5\%^{\pm 0.53}$	$98.7\%^{\pm 0.50}$	$98.1\%^{\pm 0.60}$
Closed-World Only, 3 Tab P@3				
BAPM	$88.8\%^{\pm 0.87}$	$83.4\%^{\pm 1.03}$	$87.6\%^{\pm 0.91}$	$81.2\%^{\pm 1.08}$
ARES	$96.3\%^{\pm 0.52}$	$95.2\%^{\pm 0.59}$	$95.7\%^{\pm 0.56}$	$94.9\%^{\pm 0.61}$
TMWF	$96.4\%^{\pm 0.52}$	$95.2\%^{\pm 0.59}$	$96.3\%^{\pm 0.52}$	$94.3\%^{\pm 0.64}$
CM	$96.7\%^{\pm 0.50}$	$96.0\%^{\pm 0.54}$	$96.2\%^{\pm 0.53}$	$94.6\%^{\pm 0.63}$
Open-World, 3 Tab P@3				
BAPM	$86.5\%^{\pm 0.95}$	$82.0\%^{\pm 1.06}$	$85.9\%^{\pm 0.96}$	$81.8\%^{\pm 1.07}$
ARES	$95.1\%^{\pm 0.60}$	$94.9\%^{\pm 0.61}$	$94.5\%^{\pm 0.63}$	$94.4\%^{\pm 0.64}$
TMWF	$96.0\%^{\pm 0.54}$	$95.1\%^{\pm 0.60}$	$95.7\%^{\pm 0.56}$	$94.4\%^{\pm 0.64}$
CM	$97.0\%^{\pm 0.47}$	$95.5\%^{\pm 0.57}$	$96.2\%^{\pm 0.53}$	$95.1\%^{\pm 0.60}$

Reducing network stability caused only a minor drop in accuracy. In the single-tab closed-world dataset with *dense* background traffic, a WF adversary can score an accuracy of 98.2% (using DF) with NDN and 97.9% (using DF) with ANDaNA. Compared to the *sparse* case, ANDaNA traces were slightly harder to classify correctly, though still close to perfect. The attack accuracy in the *dense* dataset consistently fell within the CI of the *sparse* one across classifiers, meaning there was minimal to no degradation in attack performance. The resilience of the classifiers in the more unstable setting likely stems from packet loss events having little impact on page loading, and timing information being more associated with webpage-specific factors rather than network conditions.

Multi-tab WF attacks succeed on NDN. The classifier performance in the multi-tab setting largely mirrors that of the single-tab. The only notable outlier is the performance of BAPM, which saw a noticable drop in P@3 between NDN and ANDaNA, from 86.5% to 82.0% with *sparse* background traffic and from 85.9% to 81.8% with *dense* background traffic. This suggests that while ANDaNA *does* offer disruption to WF attacks, modern classifiers remain sufficiently robust such that the difference is negligible.

The need for WF defenses on NDN. Despite NDN's architectural departure from TCP/IP, such as multi-path forwarding and in-network caching, existing WF classifiers retained high accuracy across all tested settings. Although world-scale NDN deployments with more websites, routers, and users would likely result in worse performance compared with this synthetic setting, our results suggest that WF is rather feasible over NDN.

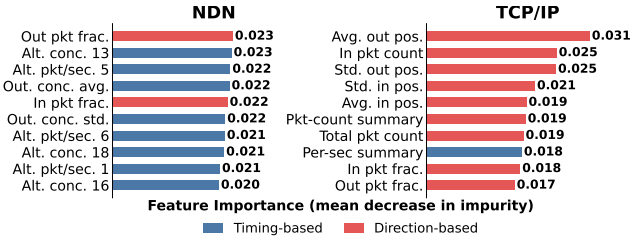

Figure 3: Closed-world feature importance for the k-FP classifier across the dense ANDaNA and TCP/IP datasets.

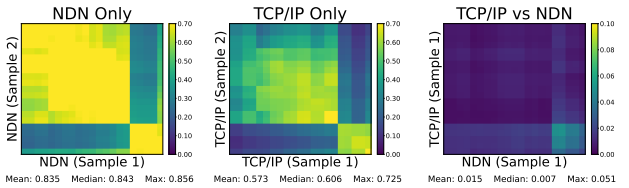

Figure 4: Closed-world CKA similarity heatmaps for Tik-Tok across the dense ANDaNA and TCP/IP datasets.

In the following sections, we examine more closely which network features these highly effective WF classifiers are using to make labeling decisions (§6.2), before examining webpages themselves to better understand what makes them easy or difficult to label (§6.3).

6.2 Inter-Network Classifier Transferability

In this section we examine whether classifiers trained on ANDaNA-NDN datasets rely on similar features and internal representations compared with Tor-TCP/IP datasets. To this end, we choose to focus our analysis in two complementary WF classifiers:

k-FP [33] represents each trace using manually engineered traffic features. Although k-FP is not a state-of-the-art deep-learning attack, its feature rankings allow us to compare the importance of packet timing, direction, ordering, and communication volume information exposed by each environment.

Tik-Tok [78], in contrast, exposes learned representations while accepting the same input in both environments. Applying this identical representation and architecture to ANDaNA-NDN and Tor-TCP/IP reduces the risk that the comparison reflects different preprocessing choices. For instance CountMamba aggregates packet events into a window-based representation [21].

Dataset construction. We create four disjoint datasets to guide our analysis: two obtained by visiting closed-world websites in NDN-ANDaNA (splitting the dense dataset used in §6.1, by keeping the same number of classes but halving the per-website samples) and two obtained by visiting closed-world websites in Tor-TCP/IP (splitting the CountMamba dataset [21]). We then train an instance of Tik-Tok [78] in each of the datasets (four in total), and an instance of the k-FP [33] attack in the first partition of each.

Manual features expose environment-specific characteristics. In Figure 3, we first analyze the manually engineered feature space used by k-FP. We rank features using Mean Decrease in Impurity (MDI), which estimates each feature's aggregate contribution to impurity reduction across the random forest. The feature rankings

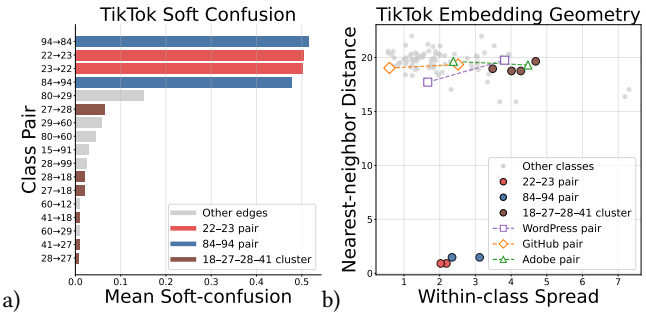

Figure 5: Soft confusion and embedding space plots of Tik-Tok on the dense ANDaNA dataset.

reveal a clear difference in the evidence used across network environments. In NDN, the highest-ranked features are dominated by timing-oriented measurements like concentration and packet-per-second features. In TCP/IP greater importance is placed on direction and ordering features like packet counts and direction.

The rankings in Figure 3 are consistent with our understanding of NDN's semantics. In NDN, traffic must follow a rigid interest-data exchange procedure and packet directions are architecturally defined. In contrast, TCP/IP packet directions are an emergent property of page loading behavior and application-layer interaction.

We hypothesize that website loading behavior typically attributed to directionality in TCP/IP is shifted towards timing-based features in NDN. For instance, a page that triggers many independent object requests during a particular phase of its load sequence can generate a dense burst of outgoing interests. Conversely, sequential dependencies can spread outgoing inter-arrival times.

Internal representations exhibit the same separation. We now examine whether feature-level divergences are also reflected in the internal representations learned by Tik-Tok. We use Centered Kernel Alignment (CKA) [52] to compare layer activations across the four trained Tik-Tok classifiers. Larger CKA values indicate more similar learned representations. The same-environment comparisons serve as positive controls: if disjoint datasets caused low CKA, then same-environment comparisons would show weak alignment.

The results in Figure 4 show that Tik-Tok models trained on disjoint NDN datasets remain strongly aligned, with mean, median, and maximum CKA values of 0.835, 0.843, and 0.856, respectively. Models trained on disjoint TCP/IP datasets also exhibit substantial same-environment alignment, with mean, median, and maximum values of 0.573, 0.606, and 0.725. In contrast, the NDN-vs-TCP/IP comparison exhibits almost no alignment, with mean, median, and maximum values of 0.015, 0.007, and 0.051, thus suggesting that the models converge on different internal representations. This separation is consistent with the implicit CC-awareness defined in §5.2: although Tik-Tok receives no explicit NDN event labels, training on NDN traces produces an environment-specific representation.

6.3 Class-Level Structure in NDN

In §6.2, we found that timing-based features are more relevant for WF in NDN compared with TCP/IP. In this section, we analyze

website-level structure, with the goal of better understanding what makes a particular website more easy or difficult to classify in NDN.

Tik-Tok's confusion is highly localized. Figure 5 shows two complementary views of Tik-Tok's learned class structure trained on the dense ANDaNA dataset. The soft-confusion chart in Fig. 5 a) highlights when Tik-Tok assigns probability mass to an incorrect class. The embedding-geometry chart in Fig. 5 b) shows class similarity in Tik-Tok's penultimate-layer using Euclidean distance.

The above plots show that Tik-Tok learns strong representations and has clear separation between almost all websites (e.g., compact embeddings and large distances from their nearest neighbor class). However, Tik-Tok is not perfect, with uncertainty concentrated in a small number of websites: classes 22–23 and 84–94 form the strongest ambiguous pairs, while classes 18–27–28–41 form a looser neighborhood. The 22–23 and 84–94 pairs are clear outliers in the soft confusion and embedding chart.

Learned similarity is not page similarity. A closer look into these ambiguous pairs shows that they do not contain obvious semantic or organizational similarities. The strongest pairs are class 22 (icloud.com) vs. class 23 (yahoo.com), and class 84 (soundcloud.com) vs. class 94 (taboola.com). These pairs do not share an apparent category, owner, or domain-level relationship. Conversely, websites that do have surface-level similarity are easily distinguished by Tik-Tok, as seen in Fig. 5 b). For instance, class 20 (wordpress.org) vs. class 48 (wordpress.com), class 14 (github.com) vs. class 61 (github.io), and class 32 (adobe.com) vs. class 86 (adobe.io).

This analysis leads to two observations. First, WF attacks in NDN struggle to label websites correctly when they have similar interest/data timing and burst structure, regardless of domain/content similarities. Second, websites that may appear related to a human remain easy to distinguish for Tik-Tok, provided their page-load process follows a unique pattern. This supports our broader interpretation from §6.2, where fingerprintability is shaped by how each page's load process is expressed through timing-oriented metadata.

6.4 Semantic Violations of WF Defenses

In the previous sections we demonstrated that off-the-shelf DL models designed for WF in TCP/IP are also very effective in NDN (§6.1), despite relying on a different representation of website loading patterns (§6.2, §6.3). In the following sections, we validate our claims from §4, by demonstrating that existing defenses cannot be ported to NDN and that our CC-versions enhance their protection.

WF defenses severely violate NDN requirements. Table 3 summarizes our findings. Across both closed-world and open-world datasets, existing WF defenses violated most of the NDN protocol requirements they could realistically affect (§4.1). For example, RegulaTor breached all three categories, while FRONT and WTF-PAD violated all except *interest timeout*, which was not applicable. The only theoretically possible but empirically unobserved violation was Tamaraw's generation of interest timeouts, which depends on the i and d values (see §4.2), and were already selected appropriately. Furthermore, the high violation percentages extending to 10,000 unmonitored websites confirms that existing WF defenses are unusable for arbitrary web content.

NDN violations are evenly distributed. Figure 6 shows heat maps with the percentage of violations incurred by each monitored website. Most defenses exhibit a relatively uniform distribution of violations across sites, further confirming that the WF defense's scheduler is the primary cause of observed NDN protocol violations, rather than website-specific patterns. The notable exception is RegulaTor, which shows several high-density regions on the heatmap, indicating some sensitivity to website-specific patterns.

Table 3: How frequently each existing WF defense violates a particular NDN architecture requirement.

Violation	Tamaraw	FRONT	WTF-PAD	RegulaTor
Dense ANDaNA, Closed-World Only				
Unsol.Data	90.00%	62.58%	14.40%	100.00%
Unsat. Interest	48.92%	21.33%	97.67%	17.83%
Timeout (20s)	0.00%	-	-	5.37%
Timeout (30s)	0.00%	-	-	1.38%
Dense ANDaNA, Open-World Only				
Unsol.Data	87.13%	62.00%	14.72%	100.00%
Unsat. Interest	49.59%	21.39%	98.69%	15.16%
Timeout (20s)	0.00%	-	-	7.50%
Timeout (30s)	0.00%	-	-	3.45%

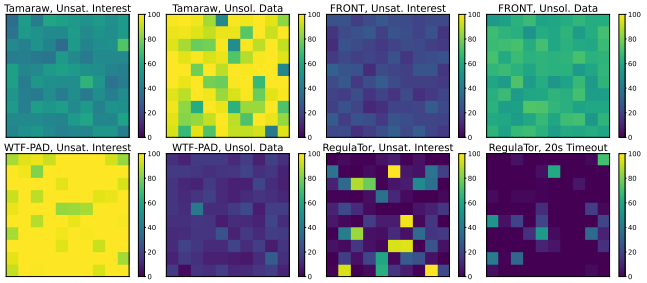

Figure 6: Violations per website for the unsolicited data, unsatisfied interest and timeout violations.

CC-variants are valid. Although not reported in Table 3, we verified that our CC adaptions for each defense do not violate any NDN protocol requirements, making them suitable for deployment.

6.5 CC-Aware Defenses in Single-tab NDN

This section addresses RQ4 (§2.4) by testing both the compatibility and efficacy of the adapted defenses. Table 4 shows the accuracy of the WF attacks in the single-tab setting after a target user (and their entry ANDaNA relay) deploy CC-aware WF defenses with *dense* background traffic. We compare the efficacy of the original version of each defense with the CC-aware versions. Although the original defenses are inoperable over NDN (see §6.4), their theoretical performance still provides a meaningful baseline. We emphasize that although the raw percentage decreases of Table 4 only indicate modest improvements, the CC-variants are working under more stringent constraints with a smaller set of actions available to protect the user (as discussed in §4).

CC-aware defenses increase or match efficacy. Across both closed-world and open-world experiments, the CC-aware WF defenses matched or outperformed their TCP/IP counterparts. CC-FRONT, CC-WTF-PAD, and CC-RegulaTor all demonstrated significant improvements, with performance outside the CI bounds of their TCP/IP variants in all 8 of 8 attack–setting combinations.

Table 4: Accuracy of WF attacks in classifying defended traces for the closed-world and open-world setting using 1 Tab.

Atk/Def	Tamaraw	(CC)	FRONT	(CC)	WTF-PAD	(CC)	RegulaTor	(CC)
Dense ANDaNA, Closed-World Only, 1 Tab Accuracy								
DF	**11.2%**$^{\pm 1.95}$	12.8%$^{\pm 2.07}$	94.8%$^{\pm 1.38}$	**92.7%**$^{\pm 1.61}$	95.5%$^{\pm 1.28}$	**93.0%**$^{\pm 1.58}$	49.7%$^{\pm 3.10}$	**36.7%**$^{\pm 2.99}$
TT	**10.3%**$^{\pm 1.88}$	11.9%$^{\pm 2.01}$	94.9%$^{\pm 1.36}$	**93.4%**$^{\pm 1.54}$	95.7%$^{\pm 1.26}$	**93.5%**$^{\pm 1.53}$	62.1%$^{\pm 3.01}$	**49.2%**$^{\pm 3.10}$
RF	**10.2%**$^{\pm 1.88}$	12.0%$^{\pm 2.01}$	95.6%$^{\pm 1.27}$	**93.0%**$^{\pm 1.58}$	96.3%$^{\pm 1.17}$	**94.4%**$^{\pm 1.43}$	69.4%$^{\pm 2.86}$	**63.0%**$^{\pm 2.99}$
CM	**11.4%**$^{\pm 1.97}$	12.1%$^{\pm 2.02}$	96.2%$^{\pm 1.19}$	**94.3%**$^{\pm 1.44}$	96.6%$^{\pm 1.12}$	**94.8%**$^{\pm 1.38}$	76.1%$^{\pm 2.64}$	**64.2%**$^{\pm 2.97}$
Dense ANDaNA, Open-World, 1 Tab Accuracy								
DF	51.5%$^{\pm 2.19}$	**50.0%**$^{\pm 2.19}$	95.3%$^{\pm 0.93}$	**94.2%**$^{\pm 1.02}$	98.0%$^{\pm 0.61}$	**97.0%**$^{\pm 0.75}$	58.8%$^{\pm 2.16}$	**55.5%**$^{\pm 2.18}$
TT	**50.3%**$^{\pm 2.19}$	51.0%$^{\pm 2.19}$	95.9%$^{\pm 0.87}$	**95.0%**$^{\pm 0.96}$	98.1%$^{\pm 0.60}$	**97.0%**$^{\pm 0.75}$	65.2%$^{\pm 2.09}$	**58.9%**$^{\pm 2.16}$
RF	51.1%$^{\pm 2.19}$	**50.3%**$^{\pm 2.19}$	96.1%$^{\pm 0.85}$	**95.2%**$^{\pm 0.94}$	98.0%$^{\pm 0.61}$	**97.2%**$^{\pm 0.72}$	68.4%$^{\pm 2.04}$	**60.3%**$^{\pm 2.14}$
CM	52.3%$^{\pm 2.19}$	**52.0%**$^{\pm 2.19}$	97.2%$^{\pm 0.72}$	**96.0%**$^{\pm 0.86}$	98.2%$^{\pm 0.58}$	**97.3%**$^{\pm 0.71}$	76.5%$^{\pm 1.86}$	**74.1%**$^{\pm 1.92}$

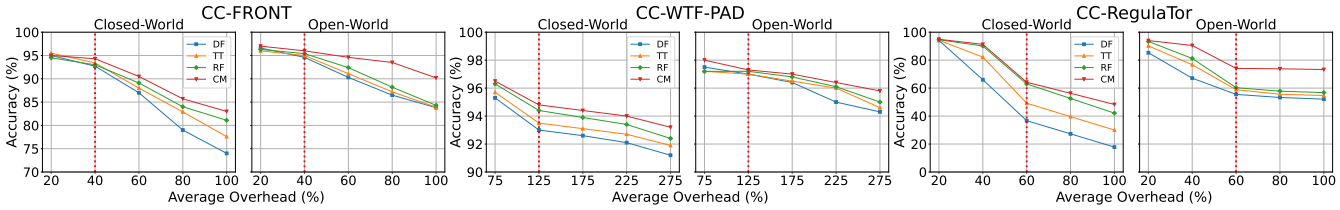

Figure 7: Overhead induced versus 1 Tab classifier accuracy. The dotted red line indicates the parameters used in Table 4.

Although Tamaraw outperformed CC-Tamaraw in 4 of 4 closed-world attacks, this increase always fell within the 95% confidence interval. Moreover, CC-Tamaraw's performance was still very high, reducing WF attack performance to just ~12.0%.

Open-world performance metrics. We primarily reported classifier accuracy for the single-tab setting in Tables 2 and 4 because accuracy and F1 score were closely linked with every attack–setting combination scoring close to 100% accuracy and/or having 100 balanced classes. In contrast, for the open-world setting, accuracy alone becomes less informative because half of the dataset contains unmonitored traces grouped into a single class. We note that the F1 score of WF attacks against Tamaraw and CC-Tamaraw ranged between just 0.05 and 0.08, confirming that the apparent ~50% accuracy was merely an artifact of the dataset's class imbalance. Similarly, the F1 score of DF and TT versus CC-RegulaTor was < 0.2, indicating better protection than accuracy suggests.

Secure web browsing on ANDaNA. Table 4 indicates that CC-Tamaraw achieves adequate protection. However, its use of a constant-rate transmission model brings steep overheads, making its deployment impractical. Excluding CC-Tamaraw, CC-RegulaTor attained the best efficacy against all attacks, suggesting that a user can prevent WF attacks on NDN by combining ANDaNA with CC-RegulaTor. We attribute CC-RegulaTor's strong performance to its dedicated inter-arrival time smoothing mechanism, which is particularly effective in NDN, where timing information is the primary driver for classifiers (as discussed in §6.2 and §6.3).

6.6 Parameter Tuning on CC-aware Defenses

Table 4 only reports baseline results using each CC-defense's default configuration parameters. To better assess each CC-defense's potential, we systematically tuned the hyperparameters for each defense and plotted the relationship between bandwidth overhead and classifier accuracy in Figure 7.

CC-FRONT delivers strong efficiency trade-offs. CC-FRONT achieves a strong balance between classifier suppression and bandwidth overhead. This relationship is not visible from the accuracy scores in Table 4 alone, but becomes clear once overhead is considered.

CC-WTF-PAD induces excessive overhead. In contrast, Figure 7 also shows that CC-WTF-PAD exhibited the highest baseline and maximum overhead. The upper bound (275%) corresponds to the cost of constant-rate defenses such as Tamaraw [21], establishing a practical ceiling for realistic deployment. This behavior mirrors prior TCP/IP evaluations of WTF-PAD [38], where attack accuracy decreased by only ~6% despite an average overhead of 54.3%. These results suggest that CC-WTF-PAD can only achieve minimal WF classifier disruption at a disproportionately high bandwidth cost.

CC-RegulaTor remains effective. Finally, the trends in Figure 7 reinforce the results in Table 4: among all CC-aware defenses, CC-RegulaTor consistently delivers the strongest performance–overhead balance, confirming it as the best defense.

6.7 CC-Aware Defenses in Multi-tab NDN

The results of WF attacks in the multi-tab setting shown in Table 5 largely mirror those from the single-tab experiments. That is, CC-FRONT, CC-WTF-PAD, and CC-RegulaTor outperformed their TCP/IP counterparts in 8 of 8 settings outside the CI and Tamaraw and CC-Tamaraw both achieved the best protection. We note that "random guessing" in the open-world 3-tab setting yields ~30% accuracy, and the F1 scores of the attacks against both Tamaraw variants were < 0.1. The overhead analysis in Figure 8 also mirrors the single-tab results, namely that CC-FRONT is somewhat effective

Table 5: P@3 of WF attacks in classifying defended traces for the closed-world and open-world setting using 3 Tabs.

Atk/Def	Tamaraw	(CC)	FRONT	(CC)	WTF-PAD	(CC)	RegulaTor	(CC)
Dense ANDaNA, Closed-World Only, 3 Tab P@3								
BAPM	**10.5%**$^{\pm 0.85}$	11.2%$^{\pm 0.87}$	74.0%$^{\pm 1.22}$	**72.1%**$^{\pm 1.24}$	86.7%$^{\pm 0.94}$	**84.1%**$^{\pm 1.01}$	21.4%$^{\pm 1.14}$	**17.9%**$^{\pm 1.06}$
ARES	**11.8%**$^{\pm 0.89}$	12.1%$^{\pm 0.90}$	93.9%$^{\pm 0.66}$	**91.6%**$^{\pm 0.77}$	93.6%$^{\pm 0.68}$	**90.3%**$^{\pm 0.82}$	55.7%$^{\pm 1.38}$	**51.5%**$^{\pm 1.39}$
TMWF	11.0%$^{\pm 0.87}$	**10.9%**$^{\pm 0.86}$	94.1%$^{\pm 0.65}$	**92.0%**$^{\pm 0.75}$	94.1%$^{\pm 0.65}$	**90.1%**$^{\pm 0.83}$	22.9%$^{\pm 1.16}$	**20.8%**$^{\pm 1.13}$
CM	10.3%$^{\pm 0.84}$	**10.0%**$^{\pm 0.83}$	94.1%$^{\pm 0.65}$	**92.3%**$^{\pm 0.74}$	93.0%$^{\pm 0.71}$	**91.0%**$^{\pm 0.79}$	58.0%$^{\pm 1.37}$	**53.5%**$^{\pm 1.38}$
Dense ANDaNA, Open-World, 3 Tab P@3								
BAPM	35.6%$^{\pm 1.33}$	**35.2%**$^{\pm 1.32}$	75.0%$^{\pm 1.20}$	**70.6%**$^{\pm 1.26}$	82.6%$^{\pm 1.05}$	**78.8%**$^{\pm 1.13}$	39.9%$^{\pm 1.36}$	**38.2%**$^{\pm 1.35}$
ARES	**36.2%**$^{\pm 1.33}$	36.5%$^{\pm 1.33}$	93.4%$^{\pm 0.69}$	**92.2%**$^{\pm 0.74}$	94.0%$^{\pm 0.66}$	**93.3%**$^{\pm 0.69}$	64.4%$^{\pm 1.33}$	**60.5%**$^{\pm 1.36}$
TMWF	35.5%$^{\pm 1.33}$	**34.9%**$^{\pm 1.32}$	92.5%$^{\pm 0.73}$	**90.3%**$^{\pm 0.82}$	94.1%$^{\pm 0.65}$	**92.4%**$^{\pm 0.73}$	43.7%$^{\pm 1.37}$	**41.5%**$^{\pm 1.37}$
CM	36.7%$^{\pm 1.34}$	**36.3%**$^{\pm 1.33}$	94.6%$^{\pm 0.63}$	**92.4%**$^{\pm 0.73}$	94.4%$^{\pm 0.64}$	**92.5%**$^{\pm 0.73}$	78.2%$^{\pm 1.14}$	**72.4%**$^{\pm 1.24}$

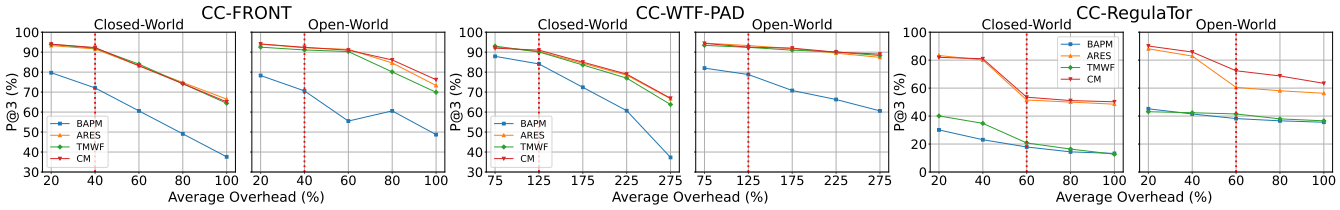

Figure 8: Overhead induced versus 3 Tab classifier P@3. The dotted red line indicates the parameters used in Table 5.

with moderate overhead, CC-WTF-PAD has extremely high bandwidth overheads for very little accuracy drop, and CC-RegulaTor provides the best protection relative to overhead.

7 Conclusion

We reported on three key results that help deepen the understanding of the applicability of traffic analysis and surveillance on NDN. First, we showed that WF attacks designed for TCP/IP are highly effective in classifying NDN traffic (§6.1), despite using a different internal representation of the data (§6.2/§6.3). Second, we showed that existing WF defenses *cannot* be transplanted directly from TCP/IP in the same manner as the attacks (§6.4). Finally, we introduced new CC-aware defense variants capable of protecting NDN users (§6.5-§6.7). Our findings also leave open several compelling research directions, which we outline below.

Explicitly CC-aware attacks. Our attacks adapt to content-centric traffic implicitly through their training data, but their inputs do not directly encode NDN-specific protocol events. An explicitly-aware CC-WF attack could use other signals available at the adversary's observation point. Candidate signals include NACK reasons, fragmentation or visible segmentation counts, repeated-Interest patterns associated with retransmissions, and cache-related delay cues. Sequence models such as Tik-Tok would need to be modified to support additional input channels or event embedding.

NDN-specific defenses. A key direction is to eschew retrofitting TCP/IP-based WF defenses and leverage NDN's architectural features. We identify three approaches that could lead to novel NDN-native defenses operating on top of encrypted NDN traffic.

One first approach to obfuscate traffic patterns is to *amplify entropy in the interest forwarding plane*. For example, routers could deliberately vary forwarding paths across equivalent name prefixes, introduce bounded forwarding delays, or probabilistically select among valid next hops to reduce the stability of timing signatures.

A second approach is to purposefully *distort interests' aggregation and caching behavior*. We hypothesize that clients or local gateways could coordinate cover interests so that multiple users' requests collapse into shared upstream behavior. Routers could also adopt caching policies that vary where content is cached, when/if cached content is served, or whether cache hits are delayed or bypassed.

Finally, a third approach is to intentionally *manipulate the content naming structure*. Defenses could introduce controlled indirection in name prefixes, ephemeral aliases for sensitive content, encrypted or partially hidden name components, or randomized segmentation.

Transferability to related settings. Our experiments focused on vanilla NDN and one PET (ANDaNA); however, similar retrieval anonymity systems like AC3N [95] and PrivICN [11] serve as a natural extension of this work. While AC3N uses the same relay routing paradigm as ANDaNA and PrivICN encrypts content names, neither obfuscates the number, direction, or timing of network packets. Although WF classifiers would likely need to be retrained to better overcome these PETs, we posit they are similarly vulnerable to traffic analysis. In turn, PETs designed to enable anonymous content publication (as opposed to our work on anonymous content retrieval) like Harpocrates [7] or Kita *et al.* [50] reflect a complimentary censorship problem that falls outside our evaluation.

Broader traffic analysis efforts targeting NDN. Additional avenues for future work include the exploration of WF attacks focusing on low data requirements [107], subpage identification [112], or early-page detection [21]. Other WF-related attacks such as video [98] and device [57] fingerprinting remain unexplored, as well as other traffic analysis tasks such as app identification [93] and malicious traffic detection [24].

Acknowledgments

This work was supported in part by NSERC under grants RGPIN-2023-03304 and CGS D-578769-2023, and by Mitacs through Globalink Research Award [IT46053]. Our experiments benefited from the use of the CrySP RIPPLE facility at the University of Waterloo and from GCP infrastructure through the Google Cloud Research Credits program.

Ethical Considerations

This work involved controlled testbed experiments and offline analyses of synthetically generated or experimentally collected network traces. It did not involve user studies, interaction with human subjects or collection of personal information. Because the work did not involve human subjects or sensitive user data, it was not submitted to an external ethics panel or IRB. We do not identify any additional ethical concerns beyond standard responsible reporting of security and privacy research.

Open Science

The source code and dataset files [102] mentioned in §3, have been made available at this link https://github.com/MichaelWrana/mnndn-scripts-new.

AI Use

The authors used generative AI-based tools to revise the text, improve flow, and correct typos, grammatical errors, and awkward phrasing. The authors also used these tools to assist with code development and documentation, including adding command-line interfaces to experimental utilities, writing scripts for file management and dataset organization, preparing VM setup and dependency-installation commands, improving inline comments, and identifying routine syntax or logic errors during debugging. We have manually verified and are responsible for the accuracy, originality, and integrity of the output of all AI-based tools.

References

[1] Alexander Afanasyev, Xiaoke Jiang, Yingdi Yu, Jiewen Tan, Yumin Xia, Allison Mankin, and Lixia Zhang. 2017. NDNS: A DNS-Like Name Service for NDN. In *IEEE International Conference on Computer Communications and Networks (ICCCN)*. 1–9.

[2] Manar Aldaoud, Dawood Al-Abri, Firdous Kausar, and Medhat Awadalla. 2024. NDNOTA: NDN One-Time Authentication. *Information* 15, 5 (2024), 289.

[3] Moreno Ambrosin, Alberto Compagno, Mauro Conti, Cesar Ghali, and Gene Tsudik. 2018. Security and Privacy Analysis of National Science Foundation Future Internet Architectures. *IEEE Communications Surveys and Tutorials* 20, 2 (2018), 1418–1442.

[4] Afia Anjum, Paul Agbaje, Arkajyoti Mitra, Emmanuel Oseghale, Ebelechukwu Nwafor, and Habeeb Olufowobi. 2024. Towards named data networking technology: Emerging applications, use cases, and challenges for secure data communication. *Future Generation Computer Systems* 151 (2024), 12–31.

[5] Guilherme B. Araújo and Leobino Sampaio. 2024. A scalable, dynamic, and secure traffic management system for vehicular named data networking applications. *Ad Hoc Networks* 158 (2024), 103476.

[6] Hitoshi Asaeda, Kazuhisa Matsuzono, Yusaku Hayamizu, Htet Htet Hlaing, and Atsushi Ooka. 2024. A Survey of Information-Centric Networking: The Quest for Innovation. *IEICE Transactions on Communications* 107, 1 (2024), 139–153.

[7] Md Washik Al Azad, Reza Tourani, Abderrahmen Mtibaa, and Spyridon Mastorakis. 2022. Harpocrates: Anonymous Data Publication in Named Data Networking. In *ACM Symposium on Access Control Models and Technologies (SACMAT)*. 79–90.

[8] Alireza Bahramali, Ardavan Bozorgi, and Amir Houmansadr. 2023. Realistic Website Fingerprinting By Augmenting Network Traces. In *ACM Conference on Computer and Communications Security (CCS)*. 1035–1049.

[9] Enkeleda Bardhi, Mauro Conti, Riccardo Lazzeretti, and Eleonora Losiouk. 2021. ICN PATTA: ICN Privacy Attack Through Traffic Analysis. In *IEEE Conference on Local Computer Networks (LCN)*. 443–446.

[10] Ahmed Benmoussa, Chaker Abdelaziz Kerrache, Nasreddine Lagraa, Spyridon Mastorakis, Abderrahmane Lakas, and Abdou El Karim Tahari. 2023. Interest Flooding Attacks in Named Data Networking: Survey of Existing Solutions, Open Issues, Requirements, and Future Directions. *Comput. Surveys* 55, 7 (2023), 139:1–139:37.

[11] César Bernardini, Samuel Marchal, Muhammad Rizwan Asghar, and Bruno Crispo. 2019. PrivICN: Privacy-preserving content retrieval in information-centric networking. *Computer Networks* 149 (2019), 13–28.

[12] Xiang Cai, Rishab Nithyanand, and Rob Johnson. 2014. CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense. In *ACM Workshop on Privacy in the Electronic Society, (WPES)*. 121–130.

[13] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. 2014. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *ACM Conference on Computer and Communications Security (CCS)*. 227–238.

[14] Pankaj Chaudhary and Neminath Hubballi. 2025. PeNCache: Popularity based cooperative caching in Named Data Networks. *Computer Networks* 257 (2025), 110995.

[15] Yeong-Sheng Chen and Wei-Cheng Chien. 2023. FIDO-Based Access Control Mechanism in Named Data Networking. In *Springer International Conference on Security and Information Technologies with AI, Internet Computing and Big-data Applications*. Springer, 263–274.

[16] Mauro Conti, Ankit Gangwal, Muhammad Hassan, Chhagan Lal, and Eleonora Losiouk. 2020. The Road Ahead for Networking: A Survey on ICN-IP Coexistence Solutions. *IEEE Communications Surveys and Tutorials* 22, 3 (2020), 2104–2129.

[17] Indrajit Das, Papiya Das, Papiya Debnath, Manash Chanda, and Subhrapratim Nath. 2025. *A Survey on NSF Future Internet Architecture (FIA) for Mobility-First (MF), Named Data Networking (NDN), NEBULA, and eXpressive Internet Architecture (XIA)*. John Wiley and Sons, Ltd, Chapter 10, 225–269.

[18] Vassilis Demiroglou, Sotiris Skaperas, Lefteris Mamatas, and Vassilis Tsaoussidis. 2024. Adaptive Multi-Protocol Communication in Smart City Networks. *IEEE Internet of Things Journal* 11, 11 (2024), 20499–20513.

[19] Xinhao Deng, Qi Li, and Ke Xu. 2024. Robust and Reliable Early-Stage Website Fingerprinting Attacks via Spatial-Temporal Distribution Analysis. In *ACM Conference on Computer and Communications Security (CCS)*. 1997–2011.

[20] Xinhao Deng, Qilei Yin, Zhuotao Liu, Xiyuan Zhao, Qi Li, Mingwei Xu, Ke Xu, and Jianping Wu. 2023. Robust Multi-tab Website Fingerprinting Attacks in the Wild. In *IEEE Symposium on Security and Privacy (S&P)*. 1005–1022.

[21] Xianwen Deng, Ruijie Zhao, Yanhao Wang, Mingwei Zhan, Zhi Xue, and Yijun Wang. 2025. Countmamba: A Generalized Website Fingerprinting Attack via Coarse-Grained Representation and Fine-Grained Prediction. In *IEEE Symposium on Security and Privacy (S&P)*. 1419–1437.

[22] Steve DiBenedetto, Paolo Gasti, Gene Tsudik, and Ersin Uzun. 2012. ANDaNA: Anonymous Named Data Networking Application. In *Network and Distributed System Security Symposium (NDSS)*. 1–18.

[23] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. 2004. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium*. 303–320.

[24] Priyanka Dodia, Mashael AlSabah, Omar Alrawi, and Tao Wang. 2022. Exposing the Rat in the Tunnel: Using Traffic Analysis for Tor-based Malware Detection. In *ACM Conference on Computer and Communications Security (CCS)*. 875–889.

[25] Shuaike Dong, Zhou Li, Di Tang, Jiongyi Chen, Menghan Sun, and Kehuan Zhang. 2020. Your Smart Home Can't Keep a Secret: Towards Automated Fingerprinting of IoT Traffic. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*. 47–59.

[26] Mustafa Ergen, Bilal Saoud, Ibraheem Shayea, Ayman A. El-Saleh, Onur Ergen, Feride Inan, and Mehmet Fatih Tüysüz. 2024. Edge computing in future wireless networks: A comprehensive evaluation and vision for 6G and beyond. *ICT Express* 10, 5 (2024), 1151–1173.

[27] Y. Fei, J. Yin, and L. Yan. 2025. Security verification framework for NDN access control. *Nature Scientific Reports* 15 (2025), 5479.

[28] Boni García, José M. del Álamo, Maurizio Leotta, and Filippo Ricca. 2024. Exploring Browser Automation: A Comparative Study of Selenium, Cypress, Puppeteer, and Playwright. In *Springer International Conference on the Quality of Information and Communications Technology (QUATIC) (Communications in Computer and Information Science, Vol. 2178)*. 142–149.

[29] Jiajun Gong and Tao Wang. 2020. Zero-delay Lightweight Defenses against Website Fingerprinting. In *USENIX Security Symposium*. 717–734.

[30] Jiajun Gong, Wuqi Zhang, Charles Zhang, and Tao Wang. 2022. Surakav: Generating Realistic Traces for a Strong Website Fingerprinting Defense. In *IEEE Symposium on Security and Privacy (S&P)*. 1558–1573.

[31] Jiajun Gong, Wuqi Zhang, Charles Zhang, and Tao Wang. 2024. WFDefProxy: Real World Implementation and Evaluation of Website Fingerprinting Defenses. *IEEE Transactions on Information Forensics and Security* 19 (2024), 1357–1371.

[32] Zhong Guan, Gang Xiong, Gaopeng Gou, Zhen Li, Mingxin Cui, and Chang Liu. 2021. BAPM: Block Attention Profiling Model for Multi-tab Website Fingerprinting Attacks on Tor. In *ACM Annual Computer Security Applications Conference (ACSAC)*. 248–259.

[33] Jamie Hayes and George Danezis. 2016. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *USENIX Security Symposium*. 1187–1203.

[34] Kai He, Shengyuan Shi, Chunxiao Yin, Hongyan Wan, and Jiaoli Shi. 2025. Private, efficient, and flexible: protecting names based on message-derived encryption in named data networking. *Cybersecurity* 8, 1 (2025), 19.

[35] Sébastien Henri, Gines Garcia-Aviles, Pablo Serrano, Albert Banchs, and Patrick Thiran. 2020. Protecting against Website Fingerprinting with Multihoming. *Privacy Enhancing Technologies Symposium (PETS)* 2020, 2 (2020), 89–110.

[36] Abdelhak Hidouri, Haifa Touati, Mohamed Hadded, Nasreddine Hajlaoui, Paul Muhlethaler, and Samia Bouzefrane. 2024. Improving NDN Resilience: A Novel Mitigation Mechanism Against Cache Pollution Attack. In *IEEE International Wireless Communications and Mobile Computing Conference (IWCMC)*. 1564–1569.

[37] Htet Htet Hlaing and Hitoshi Asaeda. 2024. ShieldDINC: Privacy-Preserving Distributed In-Network Computations with Efficient Homomorphic Encryption. In *IEEE Conference on Local Computer Networks (LCN)*. 1–6.

[38] James K. Holland and Nicholas Hopper. 2022. RegulaTor: A Straightforward Website Fingerprinting Defense. In *Privacy Enhancing Technologies Symposium (PETS)*. 344–362.

[39] Md Ariful Islam, Sharmin Sultana Mim, and Partha Mandal. 2025. NDN: A Paradigm Shift for Scalable and Secure Data Delivery. In *Springer Real-World Applications and Implementations of IoT*. 159–181.

[40] Rafiqul Islam, Claudio Savaglio, and Giancarlo Fortino. 2025. Leading Smart Environments towards the Future Internet through Name Data Networking: A survey. *Future Generation Computer Systems* 167 (2025), 107754.

[41] Mousa Tayseer Jafar, Lu-Xing Yang, Gang Li, Robin Doss, Kon Mouzakis, Rajesh Vasa, Helge Janicke, Ahmed Ibrahim, Ahmed Mohsin, Iqbal H. Sarker, Kristen Moore, Seyit Camtepe, and Diksha Goel. 2026. Mitigating malware prevalence in networks with arbitrary topologies: a Flip-It cyber game approach integrated with epidemic modeling. *Information Sciences* 726 (2026), 122753.

[42] Meiyi Jiang, Baojiang Cui, Junsong Fu, Tao Wang, Lu Yao, and Bharat K. Bhargava. 2024. RUDOLF: An Efficient and Adaptive Defense Approach Against Website Fingerprinting Attacks Based on Soft Actor-Critic Algorithm. *IEEE Transactions on Information Forensics and Security* 19 (2024), 7794–7809.

[43] Zhaoxin Jin, Tianbo Lu, Shuang Luo, and Jiaze Shang. 2023. Transformer-based Model for Multi-tab Website Fingerprinting Attack. In *ACM Conference on Computer and Communications Security (CCS)*. 1050–1064.

[44] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Díaz, and Matthew Wright. 2016. Toward an Efficient Website Fingerprinting Defense. In *European Symposium on Research in Computer Security (ESORICS)*. 27–46.

[45] Junxiao Shi (yoursunny). 2025. ndn6-tools: Tools for the NDN6 Router Network. https://github.com/yoursunny/ndn6-tools.

[46] Junxiao Shi (Yoursunny). 2025. NDNts: Named Data Networking libraries for the Modern Web. https://github.com/yoursunny/NDNts.

[47] Pushpendu Kar, Lin Chen, Weixue Sheng, Chiew Foong Kwong, and David Chieng. 2024. Advancing NDN security: Efficient identification of cache pollution attacks through rank comparison. *Internet Things* 26 (2024), 101142.

[48] Do-hyung Kim, Jun Bi, Athanasios V. Vasilakos, and Ikjun Yeom. 2017. Security of Cached Content in NDN. *IEEE Transactions on Information Forensics and Security* 12, 12 (2017), 2933–2944.

[49] Kentaro Kita. 2023. *Anonymity and Privacy in Named Data Networking*. Ph. D. Dissertation. Osaka University.

[50] Kentaro Kita, Yuki Koizumi, Toru Hasegawa, Onur Ascigil, and Ioannis Psaras. 2021. Producer Anonymity Based on Onion Routing in Named Data Networking. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 2420–2436.

[51] Katharina Kohls, David Rupprecht, Thorsten Holz, and Christina Pöpper. 2019. Lost traffic encryption: fingerprinting LTE/4G traffic on layer two. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. 249–260.

[52] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. 2019. Similarity of Neural Network Representations Revisited. In *International Conference on Machine Learning (ICML)*. 3519–3529.

[53] Matta Krishna Kumari and Nikhil Tripathi. 2025. Detecting interest flooding attacks in NDN: A probability-based event-driven approach. *Computers and Security* 148 (2025), 104124.

[54] Matta Krishna Kumari, Nikhil Tripathi, and Piyush Joshi. 2025. ProxaDyn: A Proximity-Aware Dynamic Caching Approach for Named Data Networks. *IEEE Transactions on Network Science and Engineering* 12, 3 (2025), 2360–2372.

[55] 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 Conference on Computer and Communications Security (CCS)*. 1971–1985.

[56] Bob Lantz, Brandon Heller, and Nick McKeown. 2010. A network in a laptop: rapid prototyping for software-defined networks. In *ACM Workshop on Hot Topics in Networks (HotNets)*. 19.

[57] Beibei Li, Youtong Chen, Lei Zhang, Licheng Wang, and Yanyu Cheng. 2024. HomeSentinel: Intelligent Anti-Fingerprinting for IoT Traffic in Smart Homes. *IEEE Transactions on Information Forensics and Security* 19 (2024), 4780–4793.

[58] Tianlong Li, Tian Song, and Yating Yang. 2024. iStack: A General and Stateful Name-based Protocol Stack for Named Data Networking. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 267–280.

[59] Teng Liang, Wei Huang, Xinyu Ma, Weizhe Zhang, Yu Zhang, and Beichuan Zhang. 2023. PCLive: Bringing Named Data Networking to Internet Livestreaming. In *ACM Conference on Information-Centric Networking (ICN)*. 36–45.

[60] Sai Gautam Mandapati, Chathurika Ranaweera, and Robin Doss. 2024. Advancing NDN Security for IoT: Harnessing Machine Learning to Detect Attacks. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. 53–55.

[61] Nate Mathews, James K. Holland, Se Eun Oh, Mohammad Saidur Rahman, Nicholas Hopper, and Matthew Wright. 2023. SoK: A Critical Evaluation of Efficient Website Fingerprinting Defenses. In *IEEE Symposium on Security and Privacy (S&P)*. 969–986.

[62] Wenwen Meng, Chuan Ma, Ming Ding, Chunpeng Ge, Yuwen Qian, and Tao Xiang. 2025. Beyond Single Tabs: A Transformative Few-Shot Approach to Multi-Tab Website Fingerprinting Attacks. In *ACM Web Conference (WWW)*. 1068–1077.

[63] Asya Mitseva and Andriy Panchenko. 2024. Stop, Don't Click Here Anymore: Boosting Website Fingerprinting By Considering Sets of Subpages. In *USENIX Security Symposium*. 1–19.

[64] Hamid Mozaffari, Amir Houmansadr, and Arun Venkataramani. 2019. Blocking-Resilient Communications in Information-Centric Networks Using Router Redirection. In *IEEE GLOBECOM Workshops*. 1–6.

[65] Named Data Networking Project. 2014–2025. ndn-cxx: NDN C++ Library with eXperimental eXtensions. https://github.com/named-data/ndn-cxx.

[66] Named Data Networking Project. 2024. Named Data Networking Platform. https://named-data.net/codebase/platform/.

[67] Named Data Networking Project. 2024. ndn-tools: NDN Essential Tools. https://github.com/named-data/ndn-tools.

[68] Guoshun Nan, Xiuquan Qiao, Yukai Tu, Wei Tan, Lei Guo, and Junliang Chen. 2015. Design and Implementation: the Native Web Browser and Server for Content-Centric Networking. In *ACM SIGCOMM Computer Communication Review*. 609–610.

[69] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. 2021. Defeating DNN-Based Traffic Analysis Systems in Real-Time With Blind Adversarial Perturbations. In *USENIX Security Symposium*. 2705–2722.

[70] NDN Team. 2026. NDN Interest Packet Specification. https://docs.named-data.net/NDN-packet-spec/current/interest.html.

[71] Mingshuo Nie, Dongming Chen, Dongqi Wang, and Huilin Chen. 2025. Local Optimization Policy for Link Prediction via Reinforcement Learning. *IEEE Transactions on Network Science and Engineering* 12, 2 (2025), 1224–1236.

[72] Se Eun Oh, Nate Mathews, Mohammad Saidur Rahman, Matthew Wright, and Nicholas Hopper. 2021. GANDaLF: GAN for Data-Limited Fingerprinting. In *Privacy Enhancing Technologies Symposium (PETS)*. 305–322.

[73] 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 Security and Privacy*.

[74] Victor Le Pochat, Tom van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczynski, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Network and Distributed System Security Symposium (NDSS)*. 1–15.

[75] Named Data Networking Project. 2023. mini-ndn: A Mini-NDN network emulator. https://github.com/named-data/mini-ndn.

[76] Xiuquan Qiao, Guoshun Nan, Yue Peng, Lei Guo, Jingwen Chen, Yunlei Sun, and Junliang Chen. 2015. NDNBrowser: An extended web browser for named data networking. *Journal of Network and Computer Applications* 50 (2015), 134–147.

[77] Mohammad Saidur Rahman, Mohsen Imani, Nate Mathews, and Matthew Wright. 2021. Mockingbird: Defending Against Deep-Learning-Based Website Fingerprinting Attacks With Adversarial Traces. *IEEE Transactions on Information Forensics and Security* 16 (2021), 1594–1609.

[78] 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. In *Privacy Enhancing Technologies Symposium (PETS)*. 5–24.

[79] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *Network and Distributed System Security Symposium (NDSS)*. 1–15.

[80] Najam Saqib and Sani Isnain. 2025. A Survey on Mitigation of Cache Pollution Attacks in NDN. *Acta Technica Jaurinensis* 18 (03 2025), 93–101.

[81] Mohammad Shahrul Mohd Shah, Yu-Beng Leau, Mohammed Anbar, and Ali Abdulqader Bin-Salem. 2023. Security and Integrity Attacks in Named Data Networking: A Survey. *IEEE Access* 11 (2023), 7984–8004.

[82] Shawn Shan, Arjun Nitin Bhagoji, Haitao Zheng, and Ben Y. Zhao. 2021. Patch-based Defenses against Web Fingerprinting Attacks. In *ACM Workshop on Artificial Intelligence and Security (AISec)*. 97–109.
[83] Wentao Shang, Jeff Thompson, Meki Cherkaoui, Jeff Burkey, and Lixia Zhang. 2013. NDN. JS: A javascript client library for named data networking. In *IEEE Conference on Computer Communications Workshops (INFOCOM)*. 399–404.
[84] Meng Shen, Kexin Ji, Zhenbo Gao, Qi Li, Liehuang Zhu, and Ke Xu. 2023. Subverting Website Fingerprinting Defenses with Robust Traffic Representation. In *USENIX Security Symposium*. 607–624.
[85] Anatoly Shusterman, Lachlan Kang, Yarden Haskal, Yosef Meltser, Prateek Mittal, Yossi Oren, and Yuval Yarom. 2019. Robust Website Fingerprinting Through the Cache Occupancy Channel. In *USENIX Security Symposium*. 639–656.
[86] Salvatore Signorello, Radu State, Jérôme François, and Olivier Festor. 2016. NDN.p4: Programming information-centric data-planes. In *IEEE International Conference on Network Softwarization (NetSoft)*. 384–389.
[87] Prabhjot Singh, Diogo Barradas, Tariq Elahi, and Noura Limam. 2024. Connecting the dots in the sky: Website fingerprinting in low Earth Orbit Satellite internet. In *Workshop on the Security of Space and Satellite Systems (SpaceSec)*. 1–10.
[88] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *ACM Conference on Computer and Communications Security (CCS)*. 1928–1943.
[89] 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 Conference on Computer and Communications Security (CCS)*. 1131–1148.
[90] Yafeng Song, Ming Yang, Qi Chen, Xiaodan Gu, and Yixuan Yao. 2023. Evaluating the Distinguishability of Tor Traffic over Censorship Circumvention Tools. In *IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. 917–922.
[91] Neil Spring, Ratul Mahajan, and David Wetherall. 2002. Measuring ISP topologies with Rocketfuel. *ACM SIGCOMM Computer Communication Review* 32, 4 (2002), 133–145.
[92] Junji Takemasa, Yuki Koizumi, and Toru Hasegawa. 2021. Vision: toward 10 Tbps NDN forwarding with billion prefixes by programmable switches. In *ACM Conference on Information-Centric Networking (ICN)*. 13–19.
[93] Vincent F. Taylor, Riccardo Spolaor, Mauro Conti, and Ivan Martinovic. 2018. Robust Smartphone App Identification via Encrypted Network Traffic Analysis. *IEEE Transactions on Information Forensics and Security* 13, 1 (2018), 63–78.
[94] Yannis Thomas, Nikos Fotiou, Iakovos Pittaras, and George Xylomenos. 2025. Secure and Efficient Data Spaces over Named Data Networking. In *IFIP International Conference on Networking*. 1–9.
[95] Gene Tsudik, Ersin Uzun, and Christopher A. Wood. 2016. AC3N: Anonymous communication in Content-Centric Networking. In *IEEE Annual Consumer Communications & Networking Conference*. 988–991.
[96] Kazuaki Ueda, Chikara Sasaki, and Atsushi Tagami. 2022. Towards an incremental deployment of NDN: robust bootstrapping using in-network indirection. In *ACM Conference on Information-Centric Networking (ICN)*. 171–173.
[97] Petr Velan, Milan Cermák, Pavel Celeda, and Martin Drasar. 2015. A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management* 25, 5 (2015), 355–374.
[98] Timothy Walsh, Armon Barton, and Mathias Kölsch. 2025. Improved Open-World Fingerprinting Increases Threat to Streaming Video Privacy but Realistic Scenarios Remain Difficult. In *Privacy Enhancing Technologies Symposium (PETS)*. 130–145.
[99] Tao Wang and Ian Goldberg. 2013. Improved website fingerprinting on Tor. In *ACM Workshop on Privacy in the Electronic Society (WPES)*. 201–212.
[100] Youfeng Wang and Xiuquan Qiao. 2014. Design and implementation of web browser for named data networking in Windows. In *IEEE International Conference on Communication, Security and Artificial Intelligence (ICCSAI)*. 607–612.
[101] Mingkui Wei. 2021. Domain Shadowing: Leveraging Content Delivery Networks for Robust Blocking-Resistant Communications. In *USENIX Security Symposium*. 3327–3343.
[102] Michael Wrana. 2026. Our paper codebase. In *Github*. https://github.com/MichaelWrana/mnndn-scripts-new
[103] Michael Wrana, Diogo Barradas, and N. Asokan. 2025. SoK: The Spectre of Surveillance and Censorship in Future Internet Architectures. In *Privacy Enhancing Technologies Symposium (PETS)*. 494–511.
[104] Mingshi Wu, Ali Zohaib, Zakir Durumeric, Amir Houmansadr, and Eric Wustrow. 2025. A Wall Behind A Wall: Emerging Regional Censorship in China. In *IEEE Symposium on Security and Privacy (S&P)*. 1363–1380.
[105] Qi Xia, Isaac Amankona Obiri, Jianbin Gao, Hu Xia, Xiaosong Zhang, Kwame Omono Asamoah, and Sandro Amofa. 2024. PRIDN: A Privacy Preserving Data Sharing on Named Data Networking. *IEEE Transactions on Information Forensics and Security* 19 (2024), 677–692.
[106] Xi Xiao, Xiang Zhou, Zhenyu Yang, Le Yu, Bin Zhang, Qixu Liu, and Xiapu Luo. 2024. A comprehensive analysis of website fingerprinting defenses on Tor. *Computers and Security* 136 (2024), 103577.
[107] Yi Xie, Jiahao Feng, Wenju Huang, Yixi Zhang, Xueliang Sun, Xiaochou Chen, and Xiapu Luo. 2024. Contrastive Fingerprinting: A Novel Website Fingerprinting Attack over Few-shot Traces. In *ACM Web Conference (WWW)*. 1203–1214.
[108] Diwen Xue, Reethika Ramesh, Arham Jain, Michalis Kallitsis, J. Alex Halderman, Jedidiah R. Crandall, and Roya Ensafi. 2022. OpenVPN is Open to VPN Fingerprinting. In *USENIX Security Symposium*. 483–500.
[109] Gunwoo Yoon and Byeongdo Hong. 2024. Scalable and Robust Mobile Activity Fingerprinting via Over-the-Air Control Channel in 5G Networks. *CoRR* abs/2409.12572 (2024), 1–11.
[110] Tianyuan Yu, Xinyu Ma, Varun Patil, Yekta Kocaogullar, and Lixia Zhang. 2024. Exploring the Design of Collaborative Applications via the Lens of NDN Workspace. In *IEEE International Conference on Metaverse Computing, Networking, and Applications (MetaCom)*. 89–96.
[111] Lixia Zhang, Alexander Afanasyev, Jeff Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named Data Networking. *Computer Communications Review* 44, 3 (2014), 66–73.
[112] Xiyuan Zhao, Xinhao Deng, Qi Li, Yunpeng Liu, Zhuotao Liu, Kun Sun, and Ke Xu. 2024. Towards Fine-Grained Webpage Fingerprinting at Scale. In *ACM Conference on Computer and Communications Security (CCS)*. 423–436.
[113] Jie Zhou, Jiangtao Luo, Junxia Wang, and Lianglang Deng. 2021. Cache Pollution Prevention Mechanism Based on Deep Reinforcement Learning in NDN. *Journal of Communications and Information Networks* 6, 1 (2021), 91–100.
[114] Ran Zhu, Tianlong Li, and Tian Song. 2021. iGate: NDN Gateway for Tunneling over IP World. In *IEEE International Conference on Computer Communications and Networks (ICCCN)*. 1–9.

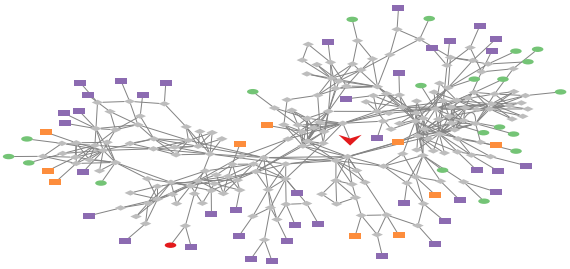

Figure 9: Ebone ISP topology with node assignments: red arrow = target user, purple rectangle = users, orange rectangle = relays, red oval = DNS server, green oval = website server.

A Network Topology

Figure 9 shows the Ebone ISP topology and associated roles assigned to each node, created following the methodology described in §3.1, and used for all data collection in §5.1 and experiments in §6.

B Protocol Violations

In §4.1 we introduced three classes of NDN protocol violations: the unsolicited data packet, unsatisfied interest packet, and interest timeout. Next, we provide contextual examples demonstrating specific scenarios where these violations occurred.

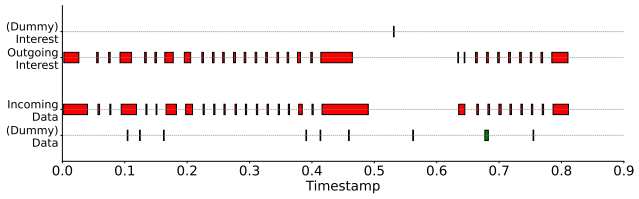

Figure 10: FRONT trace for a website access over ANDaNA. Dummy data is generated before any dummy interests.

B.1 Unsolicited Data Packets

We showcase a sample trace affected by the unsolicited data violation introduced in §4.1 in Figure 10, which depicts a traffic trace obtained by the target user on our simulation visiting an NDN website that mirrors `mail.ru` (the third most popular website obtained during crawling) over ANDaNA and defended by FRONT. Figure 10 shows the packets recorded by the adversary from the client/user's perspective with timestamps (in seconds) on the X-axis and directions on the Y-axis, using +1 for outgoing and -1 for incoming. Dummy packets have been separated with a direction of 2 and -2 for ease of illustration.

Observe that the client starts by issuing outgoing interests (i.e., real interests), and receiving the corresponding incoming data packets (i.e., real data) shortly after. However, also observe dummy data packets being generated (and sent towards the client) starting near `t=0.1`, *before* any dummy interests. This is incompatible with NDN routing, since dummy data packets would not be forwarded by any intermediate routers, and instead simply be dropped. Attempting to "piggyback" dummy data on a prior real interest is also infeasible, since fragmented data packets are each assigned a strictly increasing consecutive integer sequence number (Figure 1).

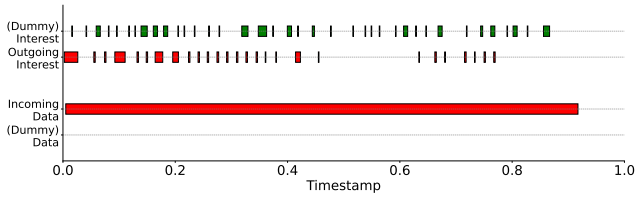

Figure 11: RegulaTor trace for a website access over ANDaNA. Dummy interests have no paired dummy data.

B.2 Unsatisfied Interest Packets

In Figure 11, we show an example of the unsatisfied interest scenario introduced in §4.1, as generated by the application of the RegulaTor defense simulator on the same website trace used in Figure 11. FRONT and WTF-PAD elicit violations in the same class.

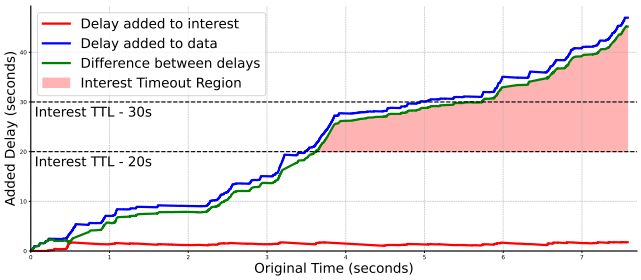

Figure 12: Trace generated by RegulaTor, showcasing excessive data packet delays. This behavior is not viable in NDN and leads to interest timeout violations.

B.3 Interest Timeout Sensitivity

In Figure 12, we illustrate the interest timeout violation introduced in §4.1 based on the application of the RegulaTor defense. The figure depicts the delay introduced by RegulaTor (Y-axis) on the sending of interest packets and the reception of the corresponding data packets, when compared to the original defended trace (X-axis). RegulaTor's scheduler delays the sending of data packet times in an unbounded fashion, while interest packet delays remain relatively steady. In this scenario, data packets would be unlikely to arrive at the destination since intermediate routers would have dropped the interest after its TTL expired. For illustration, we highlight two potential TTL values of 20s and 30s, chosen after TCP's handshake timeout values, after which interests would no longer be satisfied.