electronics

MDPI

Article

A Countermeasure Approach for Brute-Force Timing Attacks on Cache Privacy in Named Data Networking Architectures

Ertugrul Dogruluk [1,*], Joaquim Macedo [2] and Antonio Costa [2]

[1] CEiiA Centro de Engenharia e Desenvolvimento de Produto, 4450-017 Matosinhos, Portugal
[2] Centro Algoritmi, Department of Informatics, University of Minho, 4710-057 Braga, Portugal; macedo@di.uminho.pt (J.M.); costa@di.uminho.pt (A.C.)
* Correspondence: ertugrul.dogruluk@ceiia.com

Abstract: One key feature of named data networks (NDN) is supporting in-network caching to increase the content distribution for today's Internet needs. However, previously cached contents may be threatened by side-channel timing measurements/attacks. For example, one adversary can identify previously cached contents by distinguishing between uncached and cached contents from the in-network caching node, namely the edge NDN router. The attacks can be mitigated by the previously proposed methods effectively. However, these countermeasures may be against the NDN paradigm, affecting the content distribution performance. This work studied the side-channel timing attack on streaming over NDN applications and proposed a capable approach to mitigate it. Firstly, a recent side-channel timing attack, designated by brute-force, was implemented on ndnSIM using the AT&T network topology. Then, a multi-level countermeasure method, designated by detection and defense (DaD), is proposed to mitigate this attack. Simulation results showed that DaD distinguishes between legitimate and adversary nodes. During the attack, the proposed DaD multi-level approach achieved the minimum cache hit ratio (≈0.7%) compared to traditional countermeasures (≈4.1% in probabilistic and ≈3.7% in freshness) without compromising legitimate requests.

Keywords: named data networks; cache privacy; side-channel timing attacks

Citation: Dogruluk, E.; Macedo, J.; Costa, A. A Countermeasure Approach for Brute-Force Timing Attacks on Cache Privacy in Named Data Networking Architectures. *Electronics* **2022**, *11*, 1265. https://doi.org/10.3390/electronics11081265

Academic Editor: Wojciech Mazurczyk

Received: 27 February 2022
Accepted: 8 April 2022
Published: 16 April 2022

1. Introduction

Recently, Internet has been compelled to operate content production and distribution by social networking, numberless Internet-connected devices, etc. Nonetheless, such activities may not be the most suitable or applicable to be achieved over the Internet because of point-to-point communication-based architecture design.

The content-centric networks (CCN) are being proposed to overcome point-to-point communication limitations for content production and consumption. In the CCN approach, the replica of contents is cached by numberless locations (cache servers) to increase the content distribution for popular contents (e.g., YouTube, Netflix, Zoom, and social networks).

The named data networks (NDN) paradigm was presented as the latest version of CCN [1]. NDN promises named-based content and in-network caching to maximize content distribution and to increase today's content production and distribution. NDN packets do not require the content source and the destination address. Therefore, the NDN is supposed to provide enhanced privacy for the destination addresses. However, the previously cached content, in spite of its benefits, may be targeted by side-channel timing attacks to threaten the NDN privacy [2–4]. Depending on the scope of the attack, an adversary node may classify or determine the location of content consumer and producer by categorizing uncached and cached contents through the time differences from the cache.

The attacks can be mitigated by certain approaches. A typical way is to add extra time to the cache-store response for the consumer(s). For instance, statically configured countermeasure methods (delay, randomized cache, and encryption) were discussed/proposed

by works [2,4–6]. However, any additional delay or name encryption may disable (reduce availability of) the cache, which can be considered against in-networking caching-based NDN design.

This work aims to mitigate countermeasure methods efficiency issues by distinguishing legitimate and adversarial consumers. Through this work's detection approach, it is possible to execute the countermeasures only for detected adversary faces. To illustrate that, first a brute-force attack scenario is implemented on the NDN application (NDNtube). Then, the proposed approach (DaD) countermeasure method is implemented and compared with traditional countermeasure mechanisms.

Paper organization: This paper is organized as follows: Section 2 summarizes the related works on named-data-networks-based side-channel timing attacks. Section 3 introduces the NDN architecture and its features/applications. Section 4 introduces the NDN streaming applications and their features. Section 5 presents the NDN cache privacy issues. Section 6 discusses possible side-channel timing attacks on NDN streaming applications' cache privacy issues. Section 7 presents this work's main contribution which is called the detection and defense privacy approach/design. Section 8 shows primary findings for implemented network attack scenarios and results in discussion. Lastly, Section 9 draws the discussion for this work.

2. Related Works

Attack related works: The timing attacks can be deployed on various applications or services. The attack implementations can be different but the main semantics are fairly similar to each other. The works [7,8] studied how private information can be exposed by timing attacks on web applications and servers (e.g., SSL web server, HTTP). More specifically, the work [9] showed the possibility of the AES key recovery with timing attacks. The other works [2,3,6,10] studied the cache timing attacks on future network paradigms, namely on information-centric networks. However, to the best of our knowledge, these works were not focused on a specific side-channel timing attack design.

Countermeasure related works: Various countermeasures are studied or proposed to mitigate the timing attacks for information-centric networks. The work [6] developed an NDN tool called ANDāNA which uses cryptographic encryption for the named content structure to mitigate the attacks. However, this work makes each content unusable and eliminates to in-network caching feature. The work [11] proposed various delay methods for the responses to mitigate the attacks. The delay is also an efficient method to mitigate the attacks but also reduces content distribution. The work [4] presented the PrivICN tool which uses a proxy-encryption scheme to provide name and data confidentiality for information-centric networks. The PrivICN tool creates a delay with an additional computational overhead of the encrypted names. Other survey works [3,10,12,13] also studied similar countermeasure approaches to mitigate timing attacks for information-centric networks. However, to the best of our knowledge, these works were not focused on adversarial behavior or identifying the severity of the attack for ICN applications.

3. Named Data Networking Architecture

The named data networking paradigm is proposed by the works of [1,14,15]. They stated that NDN is the next generation of the IP architecture. By design, NDN names the packets instead of naming the communication endpoints, therefore changing the network semantics from delivery of packets to the destination address, to the request and caching of packets with a given name.

Figure 1 illustrates the layered hourglass designs of IP and NDN architectures. In NDN data, the name can be anything, for instance, a chunk of video, an endpoint, etc. NDN also derives/maintains the hourglass form but replaces the IP data delivery architecture with a receiver-based content retrieval architecture.

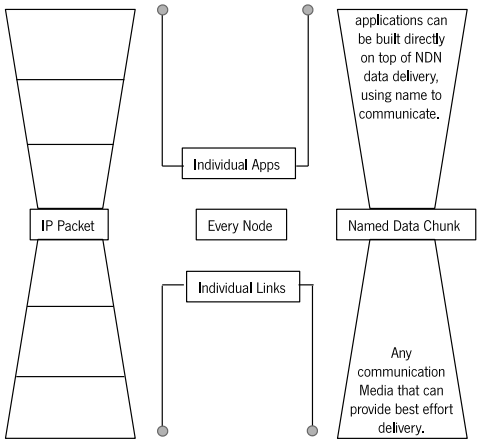


Figure 1. IP and NDN hourglass architectures (Adapted from [14]).

The hourglass model is used as a means of describing Internet design. Today's Internet hourglass architecture represents a design in layered systems that aim to support a diversity of applications and implementations. The hourglass is based on a universal network layer such as Internet Protocol (IP). This thin waist is a key enabler of Internet growth, by letting the techniques of upper and lower layers innovate independently, as described by [1]. In summary, the IP was intended to establish a communication network between destination and source addresses. However, the growth of social networks and other applications has led to the use of the Internet as a content distribution network. Therefore, using the distribution networks via the communication network is error-prone and complex to solve. NDN keeps the same Internet hourglass-shaped architecture by replacing the thin waist with named data other than communication endpoints. This semantic advances the network from delivering the packet to the given destination address to a caching data/content packet that is identified by a given name.

Figure 2 illustrates the NDN packet types. The consumer issues an interest and the producer issues a data—content packets. The interest packet express the request of consumer by giving named of the data—content (e.g., /ndn/uminho/stream-app/video-1). The data packet expresses a packet that is produced by the data/content producer. Natively, each of the data/content is signed by its producer to maintain the integrity of content and authenticate the producer, as described in the NDN security architecture [1,14].

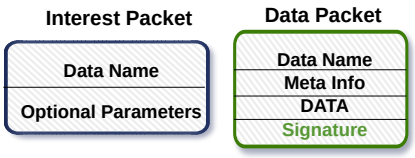


Figure 2. NDN interest (consumer) and data (producer) packets form (Adapted from [15]).

3.1. Names

The NDN name space is structured hierarchically, for instance, a video file may be produced by uminho `ndn/uminho/stream-app/video-1`, similar to the uniform resource locator (URL), with name components separated by '/' in a readable address format.

To request data by name, the consumer must build a name for the intended data. A name creation may be based on a deterministic algorithm, which lets the consumer and producer gain information for the same name. A consumer may also retrieve contents by a partially known name. This technique is also known as "*longest prefix matching*" by interest selectors which are used to specifically determine the content object. Retrieving data with a partially known name can be supported by a set of interest selectors.

In addition, any NDN application and consumer can create their namespaces, to increase mapping data and its usage of the network. The naming of the data allows

increasing mainly the functionality of data distribution, mobility, delay-tolerant networking, and multicast operations.

3.2. Data-Centric Security

To ensure the integrity of the data, all named contents (data) packets are digitally signed by their content producer. This is also known as NDN data-centric security, as discussed by [15]. It also supports data trust and allows the consumer to check the producer's public key validity. The validation can be also carried out by a hierarchical trust model, where some namespaces can be certified by private companies or entities (third parties).

NDN application layers can manage the access control to data through encryption and distribution of the keys. Additionally, using the signatures on control messages and network packets allows for securing the routing protocols.

3.3. In-Network Caching

NDN supports in-network caching features. This engages the data (content) packets to be independently retrieved or cached from the network nodes (routers).

In NDN, content store (CS) is responsible for caching named content to respond to future requests, although CS is fairly similar to today's Internet Protocol (IP) buffer memory. Nevertheless, an NDN node (router) is able to manage or rehandle the contents and network components, while IP routers cannot.

In addition, the network congestions can be resolved by NDN's content store because of the capability of retransmitting the data. For instance, in traditional IP networks, if congestion occurs between consumer–producer, and a requested data packet gets through congested links, the requested data can be dropped. This issue can be mitigated or resolved by NDN's content store because the data can remain in an intermediary node(s). Therefore, caching contents (data) in the intermediate node(s) allows the packets to be retransmitted to consumers over the second congested link. Nevertheless, in the IP network nodes, retransmission for the contents is only achieved by the producer. In this retransmission, the content packet must pass through a congested connection (link) again. On the other hand, the replacement strategies used have a significant role to improve CS performance. Relevant strategies for resource allocation and data classification are proposed in [16] and CaDaCa [17].

In summary, NDN's content store provides optimal data delivery without packet loss or congestion for the contents such as static and dynamic (e.g., real-time streaming).

3.4. Packet Forwarding and Routing

The forwarding of interest and data packets is managed by three NDN components included in all nodes: content store (CS), pending interest table (PIT), and forwarding information base (FIB), as presented by [1,14].

As mentioned previously, NDN's content store is a cache storage for the data packets, similar to today's Internet buffer router memory. FIB is used to route the interest packets which are identified by name prefix(es) table and outgoing interfaces.

Figure 3 illustrates the NDN forwarding design. In this model, when an interest packet is received by the first node, a lookup (search for the named data) is initiated by CS for the previously cached contents. In the case that the intended data has a matching at CS, it is replied to by a named content packet to the consumer. Otherwise, the router (or node) looks up the requested data (content) name in the PIT component. If a matching content name is in the PIT table, the NDN router (node) registers a consumer interface for the requested interest. Thus, the router can satisfy future incoming interest requests. Otherwise, an NDN router adds a new PIT entry and FIB can inform the sourcing interface if there are no routes to satisfy the request.

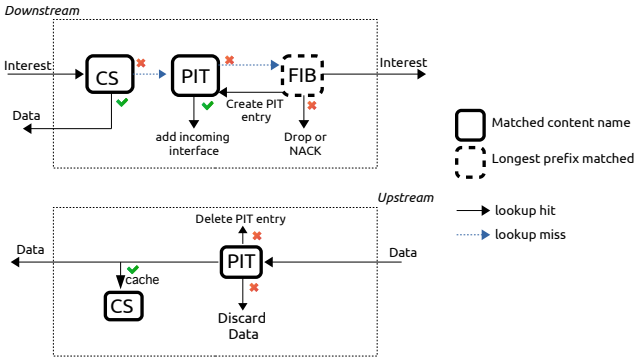


Figure 3. Named data networking forwarding (Adapted from [14]).

If lookup data are not found in the network, an interest is routed to the content producer with its FIB information. The *longest prefix match* information in FIB determines where to send it for each of the interest packets. In addition, each of the outgoing interfaces at the FIB entry is stated as a reference for the packet routing. In the end, if the name is not even found by the content producer, the issued interest becomes unsatisfied. Additionally, these packets can be used for denial of service attacks in NDN. To mitigate this, when all corresponding interest packet requests are missed and NACK (negative acknowledgments) packets occur, the forwarder can limit the requests of the NDN router, as described by [18].

Lastly, the PIT checks to match entry for arrived data packets downstream of an NDN router. When a paired entry is obtained, the named content (data) is forwarded (transmitted) to face (interface), cached, and PIT discards the entry. Additionally, PIT has the capability of discarding/removing the data packet when signature verification is failed.

3.5. Table Management

The NDN forwarding path contains three tables: FIB, PIT, and CS. The routing information base (RIB) is used to compute the FIB. All packet forwarding information is stored by FIB. The RIB is populated by NDN's routing protocol which is called the named data link state routing protocol (NLSR). Lastly, the core NDN protocol is implemented and evolved by the named data networking daemon (NFD) forwarder tool [19].

As Figure 4 illustrates, the FIB is used to forward the interests to the potential sources. The FIB is updated by the FIB management protocol, which is operated by routing a forwarding application layer.

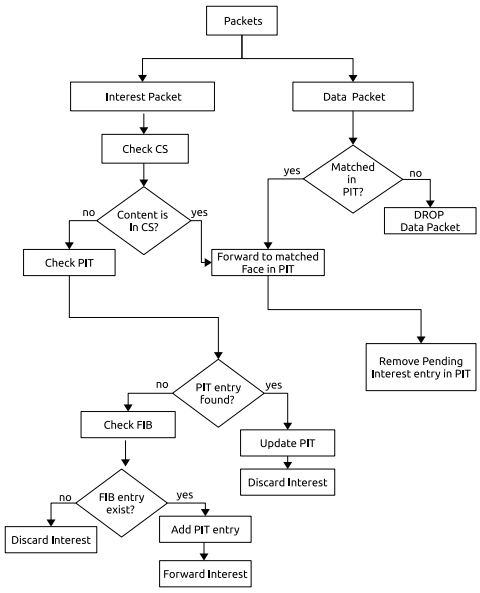


Figure 4. Single interest/data packet forwarding mechanism.

4. Streaming Applications Architecture over Named Data Networks

A streaming media user can be listening to media or watching a video in real time or prerecorded over the Internet (IP) by current streaming applications. In streaming, continuous content is delivered by the producer to the consumer. The video may be delivered to be saved in the cache for later on-demand playback by CDN caching servers. In this way, a streaming producer may handle failures and diversity, such as traffic congestion, multiple versions of an encoder, different device video resolution, etc.

The video distribution applications benefit from the NDN architecture. The live streaming tools (NDNlive, NDNvideo, and NDN-RTC) and prerecorded and live stream tools (NDNtube) were proposed by [20–23]. Instead of relying on centralized servers as it is constructed with current streaming applications, the NDN design may make the servers robust by naming the streaming packets, which can be retrieved from the network layer, independently. Through NDN, the applications fetch the streamed content by name, and the content can be delivered either by the producer or by any router's CS. That also removes the third-party application requirements of managing and locating streamed contents, as designed in NDNtube [22] and video-conferencing applications [23].

The audio conference tool (ACT) architecture over NDN, proposed by [24], takes the advantage of named data to locate the conference, the speakers, and to fetch packets from speakers. The tool also announces the conference by using a signaling protocol, which is called the session description protocol (SDP) [25]. The name is constructed as `/ndn/broadcast/conference/session/speaker-list`, for example. When the voice packet is generated by the speaker, the ACT server caches the data in SDP format, which describes the name prefix that can be used for media type, voice data, and public key locator.

Figures 5 and 6 illustrate a generic name hierarchy on NDN streaming applications. The large video content is split into video segments with their frame number. The NDN video and audio name components are presented by the following:

Routable prefix: This is used to identify the interest and forward it to the NDN network.

Application: The NDN_x is used to identify the NDN application name, such as live streaming, video, RTC, and Tube.

Stream ID: This is an identifier used to distinguish one stream from the others.

Media: This is used to identify the content type for `/video` and `/audio`.

Content: The audio and video frames are structured in the content and may be also used to identify the streaming information, such as codec H.264 and metadata.

Frame: It identifies each audio and video frame by a number.

Segment: It identifies each data frame segment.

$$\underbrace{/ndn/uminho}_{\texttt{routable prefix}}\ \underbrace{/NDN\_x}_{\texttt{application}}\ \underbrace{/stream\_1}_{\texttt{stream ID}}\ \underbrace{/video-1}_{\texttt{media}}\ \underbrace{/content}_{\texttt{content}}\ \underbrace{/frame\_num}_{\texttt{frame}}\ \underbrace{/\%00}_{\texttt{segment}}$$

Figure 5. Video streaming and audio packet format.

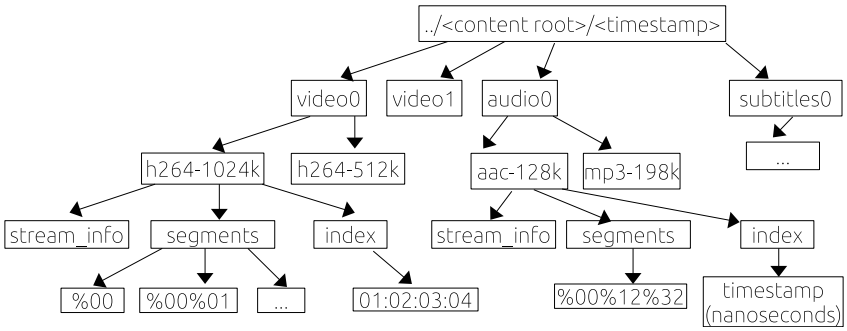


Figure 6. NDN streaming applications namespace (Adapted from [20]).

The content name may also include a content verification parameter and is identified by the metadata section of a data packet, as described by [26]. Each frame is signed by

the producer and signature carried within namespace `<stream>/key` in metadata. The metadata parameter may also include a freshness period (e.g., ≈1000 ms) of the data packet, which also defines the cache time for the data packet by its producer.

5. Cache Privacy

The digitally signed packets do not guarantee the protection against traffic analyses as described by [27]. Since each NDN node must have cache content, the privacy concerns increase for cached content. In NDN applications, cached data–contents can be objected to by an adversarial node(s). For instance, the characteristics between cached and uncached contents can be different, and can be used for identifying (or determining) the cached contents (targets) in router CS.

In NDN, the network components can store such information about the requested content/data, in order to boost the network throughput, services, and latency. These information may reflect the content about its producer, the name, consumer (requester), size, and the certificate [1–3,28]. The time differences between the cached and uncached contents may reveal private information by adversarial attacks.

To understand the adversarial attacks on NDN cache privacy, let us consider an adversary node (*Adv.*) that would like to determine whether a target/consumer (*Bob*) has requested a content/data packet (*C*—produced by *Alice*) or not. An adversary estimates the RTT (round-trip time) [$R \longleftrightarrow Adversary$] of a shared first-hop router (*R*) with Bob. Then, it issues interest for the content (*C*) and compares these two RTT values to be estimated if the targeted content (*C*)) has been cached recently or not from (*R*) (supposedly Adv. and Bob connected to the NDN router. If there are other users, this will not change the nature of the attack). Additionally, the adversary node can also determine/estimate information about the content producer (Alice).

Similarly, let us consider that Alice and Adv. share the same router or are separated at least by one router. The Adv. estimates RTT [$Alice \longleftrightarrow Adv.$] and then issues an interest request for *C*. The Adv. concludes that at least one consumer has requested *C* and is cached by the router(s) if this RTT is lower than the former RTT. Lastly, the Adv. may combine these two attack types against Bob and Alice to determine if they have exchanged packets recently or they exchange packets in real-time two-way communication, e.g., SSH or voice/video [2] (in the real-time voice/video attack, the Adv. must be a physical neighbor to the target (e.g., same shared Ethernet interface)).

Possible attack scopes on NDN streaming applications: The prerecorded and live content can be published by NDNtube, NDNvideo, and NDNlive applications. To answer live and future requests when the producer becomes offline, the video and its segments can be cached by routers. Through this approach, the load of the producer can be reduced and the video distribution maximized [20,22].

The audio/video segments are cached by the CS and each segment can be targeted by a side-channel timing attack. For instance, the video segments can be targeted by an adversary to obtain the location of the targets. The attack can be designed for a single segment by an adversary to identify the popularity of the video. In addition, the attack can be configured to monitor the cache to obtain the video types or contents by region.

6. Side-Channel Timing Attack on Streaming Applications over NDN

In this work, the attack model was specifically focused on NDN streaming application(s), where the application has a single producer and many consumers.

The privacy-oriented side-channel timing attacks/measurements can be based on information gathering from the computer implementation systems, rather than exposing software and algorithm weaknesses. The gained information, such as timing responses of the packet, can be used to identify private information by an adversary. In addition, the side-channel timing attack does not require any advanced configuration, because some information is publicly available and can be retrieved by any network consumer [29].

In NDN, whether or not the content is previously cached by CS, the consumer retrieves the packet in time which is called round trip time (RTT). It is defined as a time difference between the sent interest packet and the received data packet. Therefore, the RTT of cached content is shorter than the RTT of uncached content. In an attack, the adversary takes advantage of the RTT differences to identify the uncached and cached targets (contents) from the NDN router.

Considering all nodes must cache the data, the privacy of cached contents can be potentially targeted by side-channel timing attacks/measurements in NDN applications. The works [1–3,6,10,28,30–32] discussed that the side-channel timing attack may affect the information privacy in NDN. Depending on the scope of the attack, the adversary is mainly able to determine/identify the content name, cache (e.g., size, content popularity [33]), and the signature (e.g., certificate) through the side-channel timing attack.

6.1. Content Retrieval Time (CRT)

In this paper, content retrieval time (CRT) is presented to specify the RTT only for content retrieval. The CRT definition was used for cached and uncached contents between the edge router and the consumer (adversary included).

CRT calculation was adapted from the TCP/IP RTT estimation [34]. Figure 7 illustrates the CRT calculation from the edge router to the adversary node. In this example, the adversary sends "*Interest 1*" and retrieves "*Data 1*", and then repeats again the same "*Interest 1*". Then, the adversary analyzes the obtained CRT_1 and CRT_2 values to conclude whether "*Data 1*" was cached or uncached from the edge router. The targeted "*Data 1*" is considered cached by the NDN edge router, if the adversary obtains $CRT_1 = CRT_2$, within a very small error tolerance (note that, it is possible to have this for uncached contents from the producer with short sequential requests).

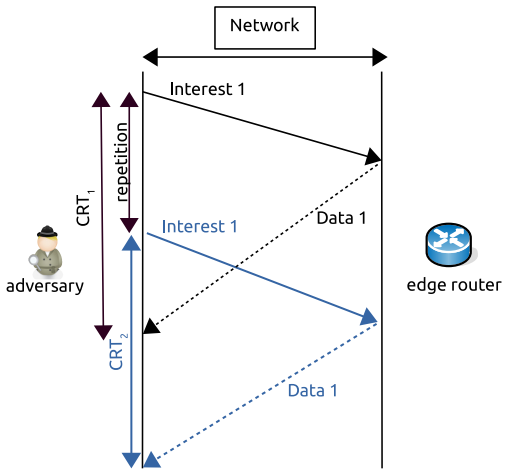


Figure 7. CRT measurements using one repetition of the same packet (Adapted from [35]).

When an interest packet is forwarded to the upstream node, the NDN router is starting a timer, which will be used to measure the CRT. When the data packet corresponding to the n interest arrives at the router, this calculates the new CRT by the equation:

$$< CRT >_n = \gamma * < CRT >_{n-1} + (1 - \gamma) * CRT_n \quad (1)$$

where n is the number of received content/data packets and $0 < \gamma < 1$. When γ is equal to $1 - \frac{1}{n}$, then the real CRT is achieved. When the γ is close to 1, then the weighted average CRT becomes not sensible to delay changes for a short time interval. When γ is close to 0, then the weighted average CRT is very sensitive to new delay changes. These CRT calculations are presented for the same interest name sequence for the adversaries [35,36].

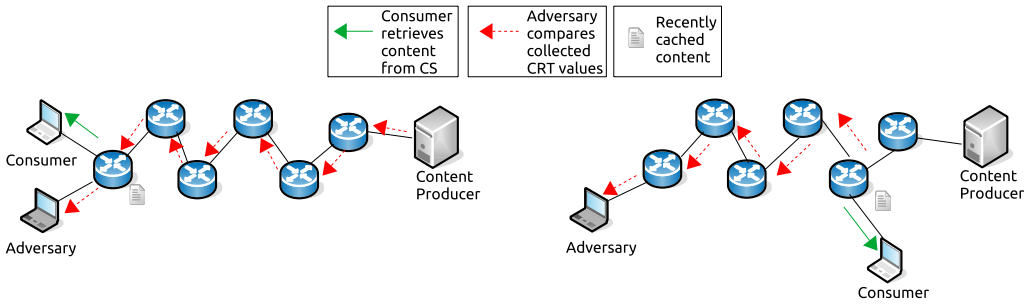
6.2. Possible Scope of Attack

In the NDN side-channel attack, the scope of the attack may vary by the intention of an adversary and be targeted to a specific streaming application (e.g., NDNtube [22], NDNlive [22], NDNvideo [20], and NDN-RTC [21]). Based on the CRT values, information privacy can be threatened by the adversary, as described by [1,2,6]. For instance, an adversary defines the targets by their content name (e.g., `/ndn-content`) or content segment (e.g., `/ndn-video/%00%12%34`) by side-channel timing attack.

The name of streamed content is defined by its producer and content itself. In streaming attack scope, an adversary may also target the name to obtain the popularity of content. Additionally, when an adversary distinguishes between cached and uncached targets, this information may be used to distinguish where the popular content has been recently cached and how long it stayed popular. This type of attack is used for content monitoring.

In a traditional attack, the sequence number is required. However, our simulation experiments showed that the attack success probability can be higher if prefix matching is allowed from the application (e.g., NDNtube or other streaming apps). For instance, instead of requesting a certain sequence number (e.g., `/ndntube/videos/video-1/%00%12%34`), the adversary can request a prefix (e.g., `/ndntube/videos/video-1`), and the router replies to the request with its segment (e.g., `/ndntube/videos/video-1/%00%12%34`).

Figure 8 illustrates two possible adversaries, based on CRT calculations. For instance, *i*. an adversary node can determine the target at the edge node (router) (Figure 8a), and *ii*. an adversary can determine the cached targets from away nodes (routers) (Figure 8b). Whenever the attack is completed to determine the locations, the attacker compares (or examines) each collected CRT metric (value), which may be used in three different main decisions for locations where the consumer's target has been cached.



(**a**) (**b**)

Figure 8. Attack designs and scopes: (**a**) Identifying closest hop cached contents. (**b**) Identifying distant hop cached contents.

For instance, *i*. a maximum CRT shows that the target has not been cached yet. *ii*. A minimum CRT shows that a target was cached by the edge node (router) recently. *iii*. A CRT between minimum and maximum shows that the target was cached by an away NDN component (e.g., router).

7. Attack and Countermeasure Designs

In this work, to meet the recent side-channel timing attack designs, a brute-force-based model was developed. In addition to this, a naïve countermeasure design is presented and proposed, which is named the detection and defense (DaD) model.

7.1. Attack Design

An adversary can improve the success of an attack by engaging multiple targets with rapid time by the brute-force attack design. Additionally, the targets can be attacked repeatedly at random intervals to increase the success of the attack [37].

Figure 9 illustrates this work's attack design which is supported by a brute-force attack process. In this attack scheme, the adversary outlines multiple predefined targets (T_n) which are the name of contents (data packets) to start the procedure of an attack. Moreover, assuming that if any of a predefined target (e.g., NDN streamed data contents) has not yet been produced (or nonexistent) by the streaming producer, the NACK packet may occur (e.g., "streamed data unavailable").

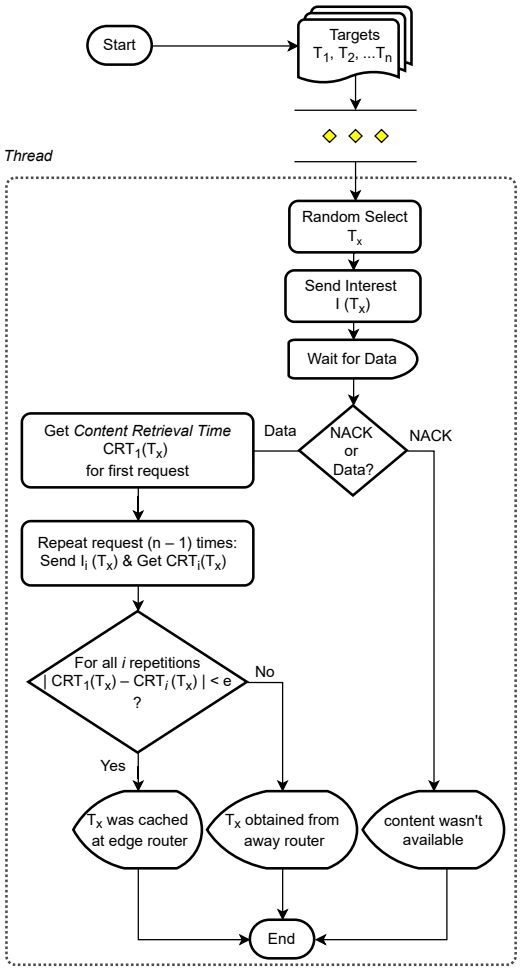


Figure 9. Attack process algorithm with the brute-force design (Adapted from [38]).

In this attack design, an adversary can repeat several attacks (depending on attack setup) to distinguish between uncached and cached targets. Each repetition presents various or matching CRT metrics to accomplish the attack decisions (e.g., the target is located or cached from the edge, neighbor, or away from locations/routers).

Attack algorithm (Figure 9) illustrates that an adversary randomly picks (or selects) the target $((T_x))$ from all predefined targets $((T_n))$. In this design, an adversary repeats the attack for each target for a minimum of two periods. In addition, the attack replication/repetition can be different through the design of the attack or purpose. The attack repetition is identified as $n-1$. After that, (CRT_i) values are collected for repetition of the target (T_x).

Whenever the attack is completed, an adversary determines (or identifies) the target locations by relying and making comparisons on CRT values. For example, if the variance accompanied by the first CRT and other CRTs is slightly small (smaller than ϵ), the CRT_i shows that a target (T_x) has just been (or recently) cached by the edge NDN node (router). Otherwise, it is cached by away NDN router(s). Furthermore, the work [39] showed that the adversaries identified geographic locations with about 90% attack success in different scenarios (including content fetch time (CFT)).

7.2. Countermeasure Design

To mitigate the side-channel timing attacks in NDN, preconfigured configurations (namely countermeasures) can be used. These can be stated as cache disabled and available methods.

The cache disabled approaches are providing a "*perfect privacy*" for the NDN cache privacy. For instance, the CS can be configured to not cache any content. An attack cannot be carried out or succeed if the content is not held in the cache [2]. For instance, the NDN anonymity tool (ANDāNA [6]) provides privacy and anonymity to the consumers. In ANDāNA, each of the requested content names is encrypted and then verified by the NDN routers.

The cache available methods are used to increase (or manipulate) the CRT value for all connected interfaces (legitimate and adversary) of the edge router. Through different CRT values, an adversary may not able to distinguish between uncached and cached targets of the NDN router. These methods are based on delay, random caching to CS as discussed by [2,10,11,40].

However, the cache disabled and available countermeasures may not be the most efficient approaches because the configuration of the NDN edge router is static. These configurations may reduce the NDN content distribution efficiency for legitimate NDN requests because of undesired transmission delays from the edge router.

To overcome or minimize countermeasures efficiency issues, detection and defense (DaD) countermeasure was proposed previously by [36,38]. In this work, the DaD was upgraded to advance adversary detection of the NDN nodes. DaD is offering an attack detection mechanism to distinguish between the adversarial and legitimate nodes to apply countermeasures to the adversary nodes.

To achieve that, the cache hit ratio (CHR) can be calculated for the "privacy sensible" contents at the NDN application layer (e.g., NFD) of the edge router. The CHR is expressed by the following equation:

$$CHR = \frac{\sum_{k=1}^{n} (total\_cache\_hits)_k}{R} \times 100\% \quad (2)$$

where n is the total number of the edge routers in the network, and R is the total number of requests received by the edge routers, which is equal to the total number of cache hits plus the total number of cache misses.

In DaD, the CHR threshold value is used. If an interface's CHR value measure is higher than the threshold, that interface is considered an adversary of the router (this scenario is considered in the edge routers).

The CHR threshold (3) parameter is calculated in DaD. A set of m requests is collected regularly during ΔT seconds. The total number of cache hits is calculated for the new set of requests, which we consider to be the ith collected set. Thus, the average CHR of this new set (chr_i) is calculated by the following equation:

$$chr_i = \frac{\sum_{k=1}^{m} CH_k}{m} \quad (3)$$

where CH_k represents the cache hit of the kth request in the new set. The CH_k is one if the kth request obtains a cache hit, and zero in case of a cache miss. Then, the new global average CHR_j is computed by the following weighted moving average equation:

$$CHR_j = (\alpha \times CHR_{j-1}) + (1 - \alpha) \times (chr_i) \quad (4)$$

where CHR_{j-1} represents the last CHR value, chr_i is the new value calculated by Equation (3), and α is a weight factor between 0 and 1. The CHR_j is very sensible to the new chr_i value if α is close to 0, and is not very sensible if α is close to 1. In DaD, α should be chosen close to 0, because an attack increases the CHR_i when it is established, and so the system can detect it quickly. For this reason, α was set close to 0 in our experimental ndnSIM scenario. The router is considered under attack if CHR_j is higher than the threshold CHR.

The CHR threshold was defined as ≈5% in NDN streaming applications (name privacy). This work used the CHR threshold to show the possibility of adversary detection, also defining attack severity.

In DaD, CHR is used *i*. to configure a countermeasure(s) when the adversary is detected, and *ii*. to determine the severity of the attack which can be used to set different countermeasures. For instance, if the attack is detected in a period (TIME) and continued in the next detection state, the attack can be considered severe. Through simulation experiences, an optimum DaD period check attack time was defined as 0.5 s for the NDNtube simulation.

Figure 10 illustrates the severity of the attack in the DaD approach. Because each of the countermeasure methods have different effects, depending on the attack severity, various countermeasure methods can be used to mitigate the attack. By doing this, the NDN content distribution is maintained while mitigating the attacks.

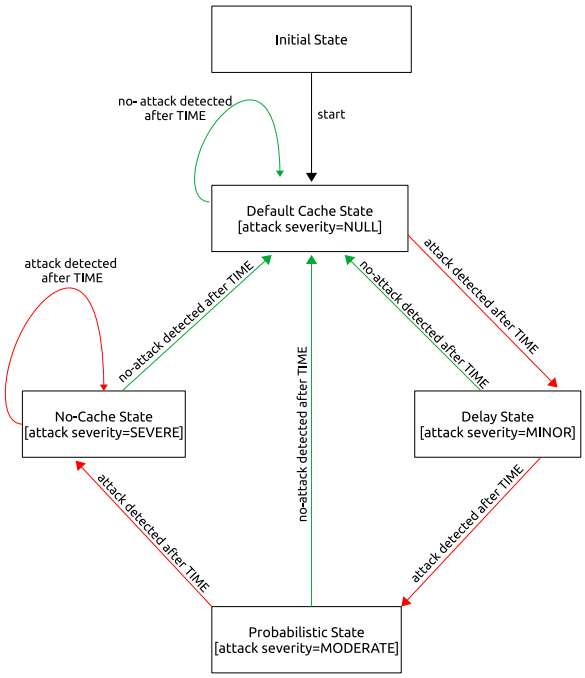


Figure 10. DaD attack states and countermeasures for the attack severity.

DaD is based on three countermeasure methods (unpredictable delay, probabilistic caching, and no-cache). The unpredictable delay is applied in the first detection state and this attack is considered as *minor*. The probabilistic caching is in the second state and this attack is considered as *moderate*. In the last attack check, if the attack continues, the router is set to no-cache for any contents, and this attack is considered as *severe*.

If the attack persists, the no-cache is applied for the next attack states. Otherwise, for a no-attack or attack–withdraw situation, the NDN router (node) sets to the default state for all consumers (e.g., LFU, LRU, or FIFO).

Table 1 shows the attack states, auxiliary process, and input parameters for the DaD approach. In this algorithm, the cacheHitTreshold and TIME parameters are previously defined by application needs. To detect an adversary for each interface (face), TIME is used as a check period. For instance, TIME can be in seconds to check faces in the NDN router.

To decide whether a router is under attack or not, the CHR is used. For instance, if a face's CHR is greater than the previously calculated cacheHitTreshold, that face is considered an adversary's face. In addition, this process is performed by getCacheHitRatio(), which is used to determine CHR for each interface of the NDN router.

Table 1. Parameters of DaD algorithm.

inputs	TIME	cacheHitTreshold
auxiliary processes	getCacheHitRatio() apply_defaultPolicy() apply_Delay() apply_Random() apply_noCache()	
attack states	ATTACK_DETECTED	delayPhase randomPhase noCachePhase
	NO_ATTACK_DETECTED	defaultPhase

The countermeasure methods are applied if DaD determined the attack (ATTACK_DETECTED). Different countermeasures are used depending on the severity of the attack. This is also identified by the attack period (TIME).

In NDNtube, DaD continuously obtains the CHR from each face and analyzes ≈50 packets (100 packets/0.5 s). If a face's CHR is over 5%, DaD sets the minor attack phase for 3 s and keeps detecting the attacks every 0.5 s. If the attack continues, DaD sets the moderate phase for another 3 s and checks the attack every 0.5 s. If the attack continues, DaD sets a severe phase and keeps it while the attack is detected.

8. Implementation and Results

To show the attack and its mitigation results, the scenario is developed using the NDN simulator (ndnSIM 2.6) [41]. The source code of the scenario can be founded at "supplementary materials".

In this scenario, an NDNtube simulation was developed to analyze adversary node behaviors during the attack. To generate network traffic, the applications were simulated using the real dataset AT&T network topology. The results are analyzed by CHR and hop-counts in *i*. brute-force attack implementation and *ii*. DaD countermeasure.

8.1. Scenario Implementation

In the streaming scenario demonstration (Figure 11), the adversary pursues the video segments cached by gateway routers previously. Therefore, the adversary (adversary-1) can knowledge the popularity of streamed contents, recently cached by the edge routers requested by streamers (streamers-1). In the attack, the adversary only probes the target (e.g., `streaming_app/videos/video-1`), then the gateway routers (edge, neighbor, and away) replies with a video segment (e.g., `streaming_app/videos/video-1/%00%12%34`) that has been cached recently. In an ideal attack, the adversary repeats each target a minimum of four times to distinguish that a target is located (or cached) by the first NDN node or the away nodes. Additionally, other adversaries are (adversary-...) attacking other streamers (streamers-..) to conclude that the streamed contents are cached (or located) by the neighbor, edge, and away routers (or nodes). In this attack, adversaries are only determined by the popularity of the streamed content by their locations.

Table 2 shows the configurations of the attack on the NDNtube-like simulation on AT&T topology. The AT&T topology has 625 nodes. A total of 156 leaves (consumers), 140 evils (adversaries), and one producer (streaming producer) were selected randomly for each one of the 10 simulation runs. The attack was applied to default CS policies (LRU, LFU, and FIFO) to evaluate the behavior of the attack (in a total of 30 runs). The adversaries have targeted the video segments named as `/ndntube/videos/video-...` and cached by gateway (edge) routers. In the NDNtube-like attack scenario, the adversaries, the streamers, and the producer were selected randomly for every scenario run. The results are obtained from each CS policy (LRU, LFU, and FIFO) and 30 simulation runs in total.

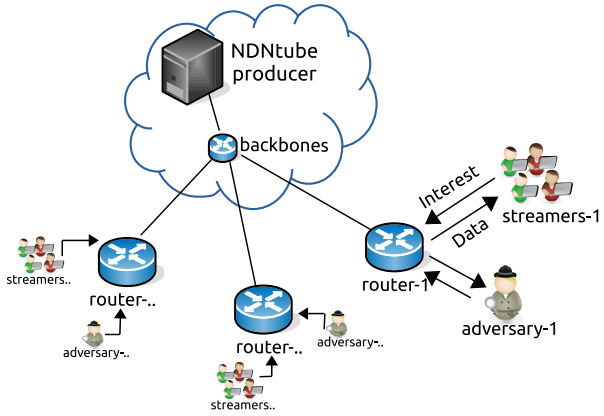


Figure 11. NDNtube attack design.

Table 2. Configurations for NDNtube attack scenario.

Network topology	AT&T
Total nodes	625
Backbones	221
Attacked edge router	108 routers
Target quantity	≈55% of total consumer nodes
Adversary quantity	≈45% of total consumer nodes
Streaming producer	/ndntube/videos/
Consumers	/ndntube/videos/...
Targets	/ndntube/videos/...
Attack repetition	4 for each target
CS policies	LRU LFU FIFO
CS size	1000 packets
CRT decisions	cached by edge node cached by neighbor node cached by away node

In this attack scenario, the adversary targets the video segments (e.g., /ndntube/videos/video-...) with a brute-force timing attack. To achieve maximized attack success, we set each request four times. Each adversary compares retrieved CRTs metrics to obtain the targets' locations and the time of the legitimate requests.

In the NDNtube simulation scenario, each of the streamed content (e.g., prerecorded video and live video) is produced by a single content producer. Each of the content is signed by the producer but it is not validated by a certificate authority. The consumers send packets (100 packets/s) at a constant rate to the content video producer. The producer virtual payload size is defined as 1024 bytes for streamed content. Note that, because NDNtube simulation is only simulated (ndnSIM) for attack and countermeasures, the audio and video decoding/encoding, compression formats (e.g., MPEG-2 and H.264) can be implemented by future studies.

Figure 12b illustrates one of the simulations run for AT&T topology. In each simulation scenario run, the routers (border nodes) were assigned randomly for the streamer producer node. In addition, the leaves nodes (namely legitimates and adversaries) were assigned randomly for the edge routers. In this network topology, the adversary nodes attack each of the gateway routers (edge) to obtain the existence of cached targets.

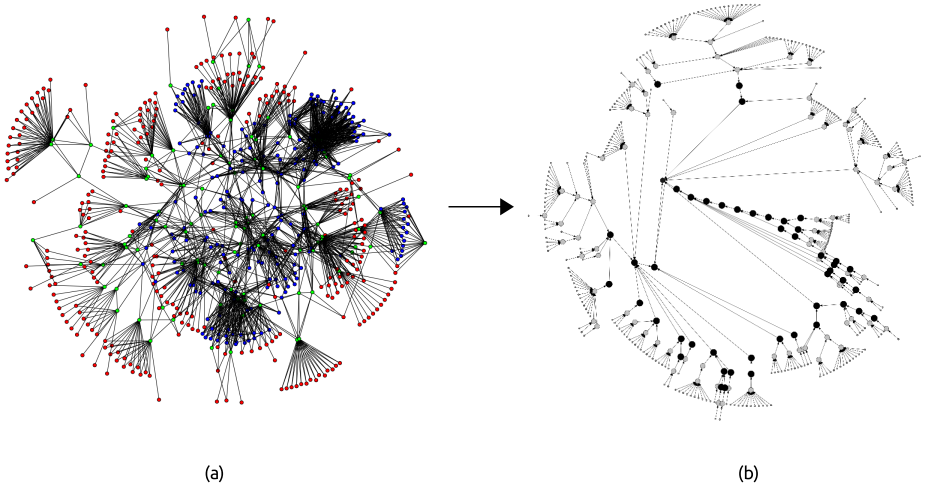


Figure 12. Real network topology conversion for NDNtube: (**a**) Raw AT&T network topology, (**b**) rocket-fuel-converted AT&T for ndnSIM.

AT&T network topology bandwidth and delays measurements (for various link types) are shown in Table 3. In this scenario, the best-route forwarding strategy is selected under different CS policies: *i*. LRU removes the least recently used streamed content segment when the CS is full. *ii*. LFU, in which the least frequently used cache block is removed whenever CS is overflowed. *iii*. FIFO, where streamed contents are evicted in the same order as they come into CS.

Table 3. Bandwidth and delays of AT&T topology.

Link Type	Delay		Bandwidth	
	Min.	**Max.**	**Min.**	**Max.**
Client–Gateway	10 ms	70 ms	1 Mbps	3 Mbps
Gateway–Backbone. Gateway–Gateway	5 ms	10 ms	10 Mbps	20 Mbps
Backbone–Backbone	5 ms	10 ms	40 Mbps	100 Mbps

8.2. Brute-Force Attack Results

The brute-force attack scenario is implemented to monitor famous video contents in the NDNtube simulation. In this attack scenario, the performance of the attacks was analyzed by different CS policies (LRU, LFU, and FIFO) on the gateway (edge) routers on AT&T topology.

Figure 13 illustrates average CHR values (or metrics) that were obtained globally from all edge routers (gw-): ≈16.4% in `nfd:LRU`, ≈15.9% in `nfd:FIFO`, and ≈18.0% in `nfd:LFU` during the attack period (≈21–35 s). In these scenarios, the attack results showed that an adversary can succeed more in the least frequently used policy because it keeps the famous contents in CS compared to other cache policies. Note that the attack success may be variously determined by the quantity of adversary, targets, network topology, and CS policy.

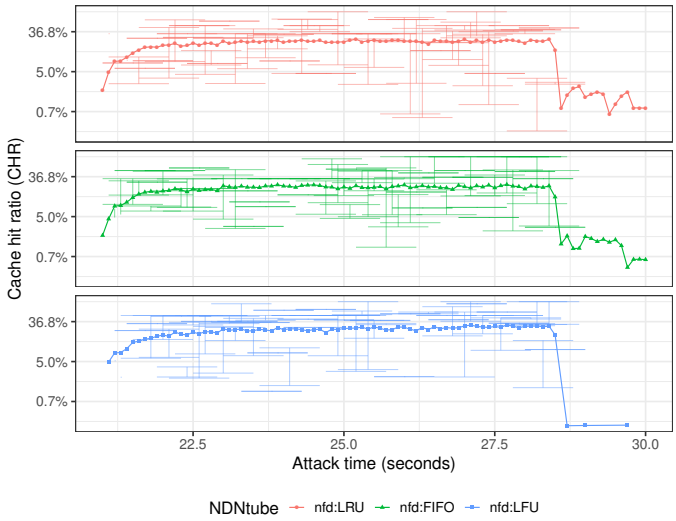


Figure 13. NDNtube brute-force attack performance on CS scenarios.

The hops between the NDN routers are identified by "hop counts". During an attack, the hop count can be decreased and used to identify the adversary's faces. Figure 14 illustrates the global hop counts for the adversary and legitimate node faces on NDNtube in LRU, FIFO, and LFU scenarios during the simulation time (0–60 s). In these results, the average, minimum, and maximum of hop counts were illustrated for the adversary and the legitimate faces, respectively. The following average hop counts were obtained: *i*. adversary faces, 1.40 in nfd:LRU, 1.95 in nfd:FIFO, and 1.52 in nfd:LFU. *ii*. legitimate faces, 3.72 in nfd:LRU, 4.15 in nfd:FIFO, and 4.15 in nfd:LFU.

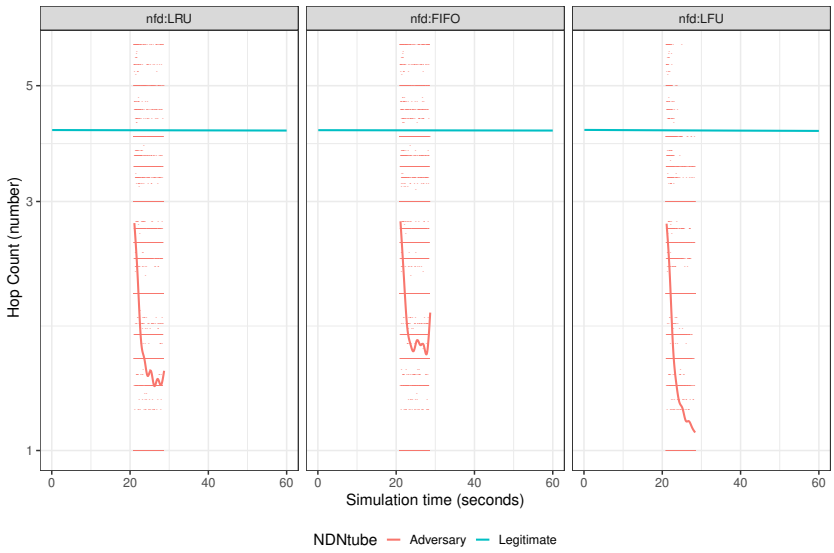


Figure 14. NDNtube global hop counts.

Figure 15 illustrates the attack findings with target locations. There are evaluated by the adversaries as the following clusters: *i*. 30.4% for edge clusters, *ii*. 18.2 % for neighbor clusters, and *iii*. 51.4% for away clusters. These results were obtained by `nfd:LRU` (default) scenario without any countermeasures applied.

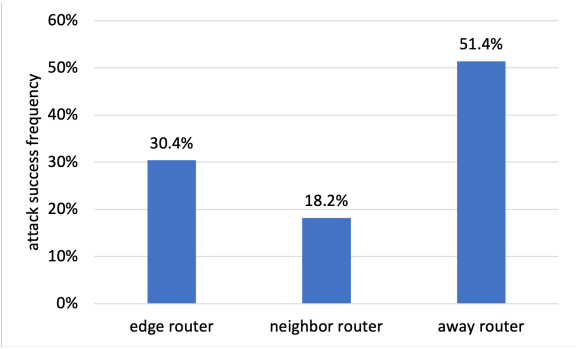


Figure 15. Determination of the target locations in AT&T topology.

8.3. Countermeasure Results

In the NDNtube simulation, the brute-force-based side-channel scenarios were implemented. To mitigate these attacks, traditional and this work's countermeasures are implemented. These are based on the following: *i*. Static probabilistic caching and freshness time, which are stated as traditional countermeasure methods. *ii*. Dynamic detection and defense configuration (DaD), which detects possible adversary detected nodes and applies the countermeasures only to these interfaces.

In NDNtube, the video segments can be also cached by the freshness period of the segment. Additionally to probabilistic caching (countermeasure to mitigate the attacks), the segments were configured by the freshness period (≈100 ms). On the other hand, the DaD threshold (CHR) is configured as 5% to detect the adversary, and the detection period is configured as 0.5 s and applies each attack phase (minor, moderate, and severe) for 3 s.

The DaD dynamically detected the attack and took countermeasure actions instead of statically configured routers to mitigate the attack. Figure 16 illustrates how DaD has dynamically mitigated the attacks in the attack period with its three phases (no-cache included). The CHR results illustrated that the `nfd:DaD` applied all countermeasures to adversarial faces because the attacks were continuous and they are considered as severe. Additionally, the statically applied (for all faces) countermeasures CHR results were obtained by the following: *i*. `nfd:probabilistic` and *ii*. `nfd:freshness` scenarios.

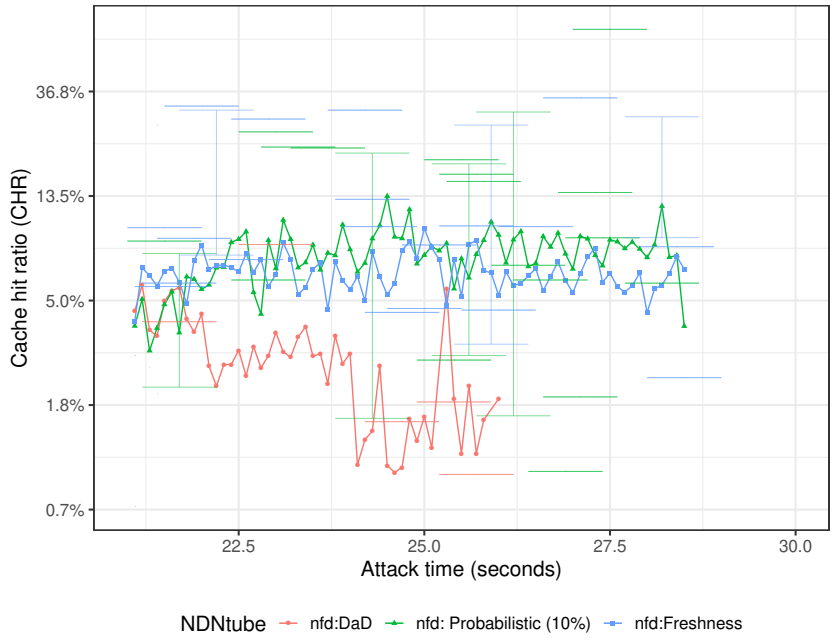


Figure 16. NDNtube attack mitigation results.

In no-countermeasure applied (Section 8.2) nfd:LRU policy scenario, the average CHR is obtained ≈16.4%. In this work, we attempted to decrease the no-countermeasure CHR value with implemented countermeasure scenarios (probabilistic, freshness, and DaD).

Firstly, the statically configured countermeasures (nfd:probabilistic ≈4.1%) and (nfd:freshness ≈3.7%) are applied and CHR is decreased compared to the no-countermeasure scenario (≈16.4%). These countermeasure results show that the attacks are mitigated. However, since all faces (including legitimate consumers) are affected by countermeasures, the content distribution efficiency is also decreased.

The nfd:DaD is using multiple countermeasures depending on attack severity. In ndn:DaD configuration, the CHR is decreased to ≈0.7% from ≈16.4%. This result showed that attack mitigation is significantly improved with DaD compared to traditional countermeasure methods. Additionally, because DaD only applies countermeasures (depending on attack severity) to possible adversary nodes, the NDN content distribution is preserved compared to traditional methods.

9. Discussion

This work attempts to mitigate the recent side-channel timing attack model (brute-force) with an efficient countermeasure method for NDN streaming applications. The attack scenarios showed that the adversaries were able to determine legitimate NDN streaming consumers by the following results: *i*. 30.4% for edge clusters, *ii*. 18.2% for neighbor clusters, and *iii*. 51.4% for away clusters.

To mitigate the attacks, DaD and traditional (statically configured) countermeasure mechanisms were evaluated. DaD achieved minimum average CHR result (≈0.7% in nfd:DaD) compared to traditional statically (≈4.1% in nfd:probabilistic and ≈3.7% in nfd:freshness).

In this work, we showed the possibility that adversarial nodes can be distinguished from legitimate consumers. However, other methods such as machine learning algorithms can be implemented to identify adversary nodes in possible future research.

Supplementary Materials: The following supporting information can be accessible/found for streaming side-channel attack scenario implementations scripts at: https://git.io/JJ35r.

Author Contributions: Conceptualization, E.D., J.M. and A.C.; methodology, E.D., J.M. and A.C.; software, E.D. and A.C.; validation, E.D. and A.C.; formal analysis, E.D., J.M. and A.C.; investigation, E.D., J.M. and A.C.; resources, E.D., J.M. and A.C.; data curation, E.D. and A.C.; writing—original draft preparation, E.D.; writing—review and editing, E.D., J.M. and A.C.; visualization, E.D. and A.C.; supervision, J.M. and A.C.; project administration, J.M. and A.C.; funding acquisition, J.M. and A.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by FCT — Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, L.; Estrin, D.; Burke, J.; Jacobson, V.; Thornton, J.D.; Smetters, D.K.; Zhang, B.; Tsudik, G.; Massey, D.; Papadopoulos, C.; et al. *Named Data Networking (NDN) Project*; Technical Report; NDN-001; 2010. Available online: https://named-data.net/wp-content/uploads/TR001ndn-proj.pdf (accessed on 12 April 2022).
2. Acs, G.; Conti, M.; Gasti, P.; Ghali, C.; Tsudik, G. Cache privacy in named-data networking. In Proceedings of the International Conference on Distributed Computing Systems, Philadelphia, PA, USA, 8–11 July 2013; pp. 41–51. [CrossRef]
3. Mohaisen, A.; Mekky, H.; Zhang, X.; Xie, H.; Kim, Y. Timing Attacks on Access Privacy in Information Centric Networks and Countermeasures. *IEEE Trans. Dependable Secur. Comput.* **2015**, *12*, 675–687. [CrossRef]
4. Bernardini, C.; Marchal, S.; Asghar, M.R.; Crispo, B. PrivICN: Privacy-preserving content retrieval in information-centric networking. *Comput. Netw.* **2019**, *149*, 13–28. [CrossRef]
5. Felten, E.W.; Schneider, M.A. Timing attacks on Web privacy. In Proceedings of the 7th ACM conference on Computer and Communications Security—CCS '00, Athens, Greece, 1–4 November 2000; pp. 25–32. [CrossRef]
6. DiBenedetto, S.; Gasti, P.; Tsudik, G.; Uzun, E. ANDaNA: Anonymous Named Data Networking Application. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 6–9 February 2011. [CrossRef]

7. Bortz, A.; Boneh, D. Exposing private information by timing web applications. In Proceedings of the 16th International Conference on World Wide Web—WWW '07, Banff, AB, Canada, 8–12 May 2007; p. 621. [CrossRef]
8. Crosby, S.A.; Wallach, D.S.; Riedi, R.H. Opportunities and Limits of Remote Timing Attacks. *ACM Trans. Inf. Syst. Secur.* **2009**, *12*, 1–29. [CrossRef]
9. Bernstein, D.J. Cache-Timing Attacks on AES. 2004; p. 37. Available online: https://cr.yp.to/antiforgery/cachetiming-20050414.pdf (accessed on 12 April 2022).
10. Chaabane, A.; De Cristofaro, E.; Kaafar, M.A.; Uzun, E. Privacy in Content-Oriented Networking: Threats and Countermeasures. *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *43*, 26–33. [CrossRef]
11. Schinzel, S. An Efficient Mitigation Method for Timing Side Channels on the Web. In Proceedings of the 2nd International Workshop on Constructive Side-Channel Analysis and Secure Design (COSADE), Darmstadt, Germany, 14 February 2011; pp. 1–6.
12. Mangili, M.; Martignon, F.; Paraboschi, S. A cache-aware mechanism to enforce confidentiality, trackability and access policy evolution in Content-Centric Networks. *Comput. Netw.* **2015**, *76*, 126–145. [CrossRef]
13. Ding, W.; Yan, Z.; Deng, R.H. A Survey on Future Internet Security Architectures. *IEEE Access* **2016**, *4*, 4374–4393. [CrossRef]
14. Zhang, L.; Afanasyev, A.; Burke, J.; Jacobson, V.; Claffy, K.; Crowley, P.; Papadopoulos, C.; Wang, L.; Zhang, B. Named Data Networking. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 66–73. [CrossRef]
15. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F.; Briggs, N.H.; Braynard, R.L. Networking Named Content. In Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies—CoNEXT '09, Rome, Italy, 1–4 December 2009; ACM Press: New York, NY, USA, 2009; Volume 30, p. 1. [CrossRef]
16. Herouala, A.T.; Kerrache, C.A.; Ziani, B.; Calafate, C.T.; Lagraa, N.; Tahari, A.E.K. Controlling the Trade-Off between Resource Efficiency and User Satisfaction in NDNs Based on Naïve Bayes Data Classification and Lagrange Method. *Future Internet* **2022**, *14*, 48. [CrossRef]
17. Herouala, A.T.; Ziani, B.; Kerrache, C.A.; el Karim Tahari, A.; Lagraa, N.; Mastorakis, S. CaDaCa: A new caching strategy in NDN using data categorization. *Multimed. Syst.* **2022**. [CrossRef]
18. Yi, C.; Afanasyev, A.; Wang, L.; Zhang, B.; Zhang, L. Adaptive forwarding in named data networking. *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *42*, 62. [CrossRef]
19. Afanasyev, A.; Shi, J. Zhang, B.; Zhang, L.; Moiseenko, I.; Afanasyev, A.; Shi, J.; Yu, Y.; Shang, W.; Li, Y.; et al. *NFD Developer's Guide*; Technical Report; NDN-0021; 2018. Available online: https://www.researchgate.net/publication/325670481_NFD_Developer%27s_Guide?channel=doi&linkId=5b1cb0d30f7e9b68b42b0ba4&showFulltext=true (accessed on 12 April 2022).
20. Kulinski, D.; Burke, J. *NDNVideo: Live and Pre-Recorded Streaming Using NDN*; Technical Report; NDN-0007; pp. 1–17. Available online: https://named-data.net/publications/techreports/trstreaming/ (accessed on 12 April 2022).
21. Gusev, P.; Burke, J. NDN-RTC: Real-Time Videoconferencing over Named Data Networking. In Proceedings of the 2nd International Conference on Information-Centric Networking—ICN '15, San Francisco, CA, USA, 30 September–2 October 2015; pp. 117–126. [CrossRef]
22. Wang, L. *NDNlive and NDNtube: Live and Prerecorded Video Streaming over NDN*; Technical Report; NDN-0031; pp. 1–10. Available online: https://named-data.net/publications/techreports/ndn-0031-1-ndnlive-ndntube/ (accessed on 12 April 2022).
23. Gusev, P.; Wang, Z.; Burke, J.; Zhang, L.; Yoneda, T.; Ohnishi, R.; Muramoto, E. Real-Time Streaming Data Delivery over Named Data Networking. *IEICE Trans. Commun.* **2016**, *E99.B*, 974–991. [CrossRef]
24. Zhu, Z.; Wang, S.; Yang, X.; Jacobson, V.; Zhang, L. ACT: Audio Conference Tool Over Named Data Networking. In Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking, Toronto, ON, Canada, 19 August 2011; Volume 11, p. 68. [CrossRef]
25. Handley, M.; Jacobson, V.; Perkins, C. *SDP: Session Description Protocol*; RFC 4566; Technical Report; The Internet Society: Reston, VA, USA, 2006. [CrossRef]
26. Mastorakis, S.; Gusev, P.; Afanasyev, A.; Zhang, L. Real-Time Data Retrieval in Named Data Networking. In Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, China, 15–17 August 2018; pp. 61–66. [CrossRef]
27. Ambrosin, M.; Compagno, A.; Conti, M.; Ghali, C.; Tsudik, G. Security and Privacy Analysis of National Science Foundation Future Internet Architectures. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 1418–1442. [CrossRef]
28. Compagno, A.; Conti, M.; Losiouk, E.; Tsudik, G.; Valle, S. A Proactive Cache Privacy Attack on NDN. In Proceedings of the NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–7. [CrossRef]
29. Kocher, P.C. *Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems*; Springer: Berlin, Germany, 1996; pp. 104–113. [CrossRef]
30. Tobias Lauinger. Security & Scalability of Content-Centric Networking. Master's Thesis, TU Darmstadt, Darmstadt, Germany and Eurécom, Sophia-Antipolis, France, 2010.
31. Lauinger, T.; Laoutaris, N.; Rodriguez, P.; Strufe, T.; Biersack, E.; Kirda, E. Privacy risks in named data networking: What is the cost of performance? *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *42*, 54–57. [CrossRef]
32. Lauinger, T.; Laoutaris, N.; Rodriguez, P.; Strufe, T.; Biersack, E.; Kirda, E. *Privacy Implications of Ubiquitous Caching in Named Data Networking Architectures*; Technical Report; 2012. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.4413&rep=rep1&type=pdf (accessed on 26 February 2022).

33. Naeem; Nor; Hassan; Kim. Compound Popular Content Caching Strategy in Named Data Networking. *Electronics* **2019**, *8*, 771. [CrossRef]
34. Douglas E. Comer. *Internetworking with TCP/IP*; Prentice-Hall: Hoboken, NJ, USA, 2000.
35. Dogruluk, E.; Costa, A.; Macedo, J. Identifying Previously Requested Content by Side-Channel Timing Attack in NDN. In *Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2018; Volume 878, pp. 33–46. [CrossRef]
36. Dogruluk, E.; Costa, A.; Macedo, J. A Detection and Defense Approach for Content Privacy in Named Data Network. In Proceedings of the 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Canary Islands, Spain, 24–26 June 2019; pp. 1–5. [CrossRef]
37. Pham, T.N.D.; Yeo, C.K.; Yanai, N.; Fujiwara, T. Detecting flooding attack and accommodating burst traffic in delay-tolerant networks. *IEEE Trans. Veh. Technol.* **2018**, *67*, 795–808. [CrossRef]
38. Dogruluk, E.; Gama, O.; Costa, A.D.; Macedo, J. Public Key Certificate Privacy in VoNDN: Voice Over Named Data Networks. *IEEE Access* **2020**, *8*, 145803–145823. [CrossRef]
39. Compagno, A.; Conti, M.; Gasti, P.; Mancini, L.V.; Tsudik, G. Violating Consumer Anonymity: Geo-Locating Nodes in Named Data Networking. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Applied Cryptography and Network Security, ACNS 2015; Springer: Cham, Switzerland, 2015; Volume 9092, pp. 243–262. [CrossRef]
40. Psaras, I.; Chai, W.K.; Pavlou, G. Probabilistic in-network caching for information-centric networks. In Proceedings of the Second Edition of the ICN Workshop on Information-Centric Networking—ICN '12, Helsinki, Finland, 17 August 2012; ACM Press: New York, NY, USA, 2012; p. 55. [CrossRef]
41. Mastorakis, S.; Afanasyev, A.; Moiseenko, I.; Zhang, L. *ndnSIM 2: An Updated NDN Simulator for NS-3*; Technical Report; NDN-0028; 2016. Available online: https://named-data.net/publications/techreports/ndn-0028-2-ndnsim-v2/ (accessed on 12 April 2022).