A Survey on Security Attacks and Intrusion Detection Mechanisms in Named Data Networking

Abdelhak Hidouri, Nasreddine Hajlaoui, Haifa Touati, Mohamed Hadded, Paul Muhlethaler

computers

MDPI

Article

A Survey on Security Attacks and Intrusion Detection Mechanisms in Named Data Networking

Abdelhak Hidouri [1,2], Nasreddine Hajlaoui [1,3], Haifa Touati [1,*], Mohamed Hadded [4] and Paul Muhlethaler [5]

[1] Hatem Bettaher IResCoMath Research Lab, University of Gabes, , Tunisia; abdelhakhdr@gmail.com; nasreddine.hajlaoui@fsg.rnu.tn
[2] National School of Computer Science (ENSI), University of Manouba, , Tunisia
[3] Unit of Scientific Research, Applied College, Qassim University, , Saudi Arabia
[4] IRT SystemX, Palaiseau, France; mohamed.elhadad@irt-systemx.fr
[5] National Institute for Research in Digital Science and Technology (INRIA), , France; paul.muhlethaler@inria.fr
* Correspondence: haifa.touati@cristal.rnu.tn

Abstract: Despite the highly secure content sharing and the optimized forwarding mechanism, the content delivery in a Named Data Network (NDN) still suffers from numerous vulnerabilities that can be exploited to reduce the efficiency of such architecture. Malicious attacks in NDN have become more sophisticated and the foremost challenge is to identify unknown and obfuscated malware, as the malware authors use different evasion techniques for information concealing to prevent detection by an Intrusion Detection System (IDS). For the most part, NDN faces immense negative impacts from attacks such as Cache Pollution Attacks (CPA), Cache Privacy Attacks, Cache Poisoning Attacks, and Interest Flooding Attacks (IFA), that target different security components, including availability, integrity, and confidentiality. This poses a critical challenge to the design of IDS in NDN. This paper provides the latest taxonomy, together with a review of the significant research works on IDSs up to the present time, and a classification of the proposed systems according to the taxonomy. It provides a structured and comprehensive overview of the existing IDSs so that a researcher can create an even better mechanism for the previously mentioned attacks. This paper discusses the limits of the techniques applied to design IDSs with recent findings that can be further exploited in order to optimize those detection and mitigation mechanisms.

Keywords: Named Data Networking (NDN); taxonomy of NDN attacks; NDN security; countermeasures; Intrusion Detection Systems (IDS)

Citation: Hidouri, A.; Hajlaoui, N.; Touati, H.; Hadded, M.; Muhlethaler, P. A Survey on Security Attacks and Intrusion Detection Mechanisms in Named Data Networking. *Computers* **2022**, *1*, 0. https://doi.org/

Academic Editor: Firstname Lastname

Received: 23 November 2022
Accepted: 7 December 2022
Published:

1. Introduction

Internet architecture was developed in the late 1970s, and its main goal was to insure communication between a limited number of equipment. As soon as the release of the World Wide Web, the original design of the Internet faced a lot of requirements such as mobility, security, and IP address scarcity. All these requirements pose many serious challenges to the traditional TCP/IP architecture. Among these challenges, is the inefficiency of the security model deployed in the TCP/IP architecture with the increasing number of vulnerabilities and attacks that are recorded every day, such as Denial-of-Service attacks (DoS), Distributed Denial-of-Service attacks (DDoS), relay attack, traffic analysis attacks, flooding attacks, etc. Moreover, massive content distribution has changed data communication in recent years.

Today's Internet is mostly characterized by the consumption of multimedia material. Users produce and consume a huge amount of data every day in an unregulated and distributed manner, and it is clear that network usage has shifted significantly to become dominated by content distribution and retrieval, while the underlying infrastructure is still based on the interconnection of hosts by means of their IP addresses. This has motivated researchers to reconsider the design of the Internet architecture.

In this context, many solutions have been suggested to deal with the vast amount of data traffic, such as Content Delivery Networks (CDNs) [1], Peer to Peer (P2P) [2] and Distributed Database (DDB) [3]. Most of these solutions are based on the deployment of dedicated servers situated in areas close to consumers to hold content replicas and optimize the download experience for consumers. Nonetheless, the widespread adoption of these solutions is far from being an ideal solution for the content delivery issue. Although, this requires ISP cooperation and DNS configurations. Furthermore, they impose significant operational and capital costs that can only be afforded by a small number of large commercial companies. Moreover, these solutions are still vulnerable to multiple security issues.

Recently, a new research direction in networking, called Information-Centric Networks (ICN), has been proposed to deal with massive content distribution. The philosophy of ICN is to focus on the content itself and not on its location. Several ICN architectures have been proposed in the literature, mainly in the United States and Europe, such as TRIAD [4], DONA [5], PSIRP and its successor PURSUIT [6], CCN [7] and NDN [8]. Among all these projects, the NDN architecture seems to be the most promising.

Unlike the TCP/IP architecture, which focuses on end-to-end points for communication, the NDN architecture focuses on the name of the content itself and pushes a signature on top of it. In NDN, each content is identified by a URI-like name and can be cached in intermediate NDN routers to serve subsequent requests for the same content. Moreover, NDN guarantees the validity of the content and each piece of content is self-signed by its provider. The "self-certified content", that is embedding the data with signature metadata, provides a direct verification of the content and its holder. This ensures the integrity and the authenticity of the content and its associated hosts and makes the NDN architecture resistant to TCP/IP traditional attacks, such as DoS attacks, DDoS attacks, or even Packet Analysis attacks.

Despite the high level of security offered by NDN and the various new features that help to improve different security and private aspects, this architecture remains vulnerable to several new security issues. The lack of addresses, for example, makes it difficult to identify the source and destination entities of a communication at any point in the network. Nonetheless, the usage of human-readable names and the presence of caches may expose the data content and make it a target for several security attacks.

1.1. Related Surveys

In the last few years, several surveys have been published regarding the security aspect of the NDN architecture.A detailed summary of existing surveys is shown in Table 1. The main criteria for choosing these surveys are as follows: (1) We used specific search words, namely "Security Attacks in NDN", "NDN IDSs", etc, to select surveys papers that have the same topic as our survey; (2) We select the surveys that discussed the same type of attacks mentioned in our survey; (3) Finally, we consider the novelty of the surveys: we selected recent surveys published from 2017 until 2022.

Respecting the chronology of releasing each survey, the first recent survey [9] introduced by Reza Tourani et al. focused mainly on DDoS attack scenarios over NDN, and it classified the mitigation mechanisms into three main categories: Rate Limiting, Statistical Modelling, Other Countermeasures. The survey introduced each attack well, but it briefly discussed some recent relevant works. In addition, the authors' resources are limited by outdated research challenges that have been resolved by the recent updates of the NDN packets specifications.

Naveen Kumar et al. [10], have gone into a further investigation of several attacks that can impact the NDN performance with relevant solutions that gave good results against a Cache Pollution Attack (CPA), Interest Flooding Attack (IFA), Cache Poisoning Attack and Cache Privacy Attack. Despite the good analysis, the survey provides multiple outdated mitigation mechanisms due to the changes of the NDN packet specification, its exposure to the malicious consumers, the high CPU usage, and the exhaustion of the space storage in the NDN routers. The survey also does not mention recent highly relevant mechanisms.

On the other hand, Hyeonseung Im et al. [11], focused mainly on studying the impact of Cache Poisoning Attacks on the NDN network and introduced the mitigation mechanisms based on basic categories such as Lightweight Verification, Selective Verification, and Detouring category. The survey misses a lot of relevant works. Moreover, The authors did not mention the drawbacks of each solution.

G. Arulkumaran and N. R. Rajalakshmi [12] provided an overview of the attacks in NDN based on the physical and data link layer, network layer, strategy layer and application layer. The authors did not mention in their research the impact of each attack on the network. Moreover, the authors briefly discussed the pre-existed mitigation mechanisms, where a wide range of mitigation mechanisms can be investigated even further. The authors also did not take into consideration the updates of the NDN packets' specifications.

IFA is the main focus of the survey introduced by Ahmed Benmoussa et al. [13]. In this survey, the authors introduced the major impact, statistical detection and mitigation mechanisms of the previously mentioned attack. The authors did not cover most of the recent mitigation mechanisms, especially the ones based on Artificial Intelligence (AI) techniques. Ren-Ting Lee et al. [14] studied several mitigation mechanisms for the IFA and classified them into nine main categories: Rate-Based, Rule-Based, Attribute-Based, Charging and rewarding, Machine Learning (ML)/Deep Learning (DL), Game Theory, Cryptography, Architecture Modification, and Wavelet Analysis. In this survey, the authors focus on the theoretical explanation of each mitigation mechanism without making an efficient comparison of the presented IDSs or identifying the limits of each solution.

Mohammed Shahrul et al. [15] introduced the attacks that can affect the NDN naming and forwarding, and slightly discussed the impact of the attacks on the NDN architecture. The survey did not cover the most recent solutions and focus on specifying the advantage of each mitigation mechanism.

The authors in [16], focused on the Basic Interest Flooding Attack (BIFA), the proposed mitigation solutions and their limitations. However, this survey did not cover recent categories of the IFA attack such as the Collusive Interest Flooding Attack (CIFA), combined BIFA and CIFA (I-CIFA) and Smart Collaborative Attack in NDN (SCAN). Moreover, the authors did not discuss the other attacks that can target the NDN security.

In [17], the authors discussed how NDN can be used as a better architecture in an IoT network, specifically for the Disaster Management use case. The authors focus on the investigation of the NDN security, more precisely the IFA; however, they did not take into consideration multiple categories of the IFA attack. In addition, the study did not consider other different attacks that can target the IoT devices that use NDN such as cache pollution attack, cache poisoning attack and cache privacy attack.

In the same baseline, the surveys suggested in [18,19] have a similar gap. In fact, these surveys slightly present the attacks that can target the NDN-IoT devices and did not cover either a large scope of the recent attacks and solutions in the NDN architecture, nor the limitations of the state-of-the-art solutions that they presented.

Table 1. State-of-the-art surveys and their limits.

Survey Reference	Topic	Drawbacks	Year of Release
[9]	Security, privacy, and access control in NDN	• Briefly discussed recent works • Limited resources • Outdated research challenges	2017
[18]	Leveraging NDN for Fragmented Networks in Smart Metropolitan Cities	• IFA countermeasures were not mentioned.	2018
[10]	Security attacks in Named data networking	• Several mechanisms outdated mitigation mechanism • Does not mention recent highly relevant mechanisms	2019
[12]	NDN design and security attacks	• Briefly discussed the pre-existed mitigation mechanisms • Did not take in consideration the updates on the packet specifications	2019
[11]	An Overview of content poisoning in NDN	• Limited state-of-the-art resources • Focused on the outdated solution	2020
[17]	NDN for Efficient IoT-based disaster management in a Smart Campus	• Misses many recent relevant research. • Did not cover most of the recent mitigation mechanism	2020

Table 1. *Cont.*

Survey Reference	Topic	Drawbacks	Year of Release
[14]	IFA in NDA and security challenges	• Did not make an efficient comparison with the presented IDSs • Did not mention the limit of each IDS	2021
[15]	NDN Future Security Challenges	• Did not cover the most recent solutions • The comparison given is very limited	2021
[13]	IFA in NDN	• Did not cover most of the recent mitigation mechanism	2022
[16]	IFA and its countermeasures in NDN	• Briefly discuss other attacks in NDN architecture.	2022
[19]	Selective content retrieval in ICN	• Does not compare the cited solutions in depth. • There are very few solutions that have been cited. • Briefly discuss recent NDN attacks.	2022
Our Survey	Security attacks and intrusion detection mechanisms in NDN	-	2022

1.2. Motivation and Goal of the Paper

Most of the survey papers mentioned above have discussed several attacks that could impact the NDN operation and presented the proposed IDSs to mitigate them. However, they still suffer from multiple drawbacks, such as not covering all potential attacks that target the NDN architecture and/or not covering recent detection mechanisms presented nowadays in the state-of-the-art technologies. Those limitations motivate us to further investigate a larger range of security threats in NDN with a deep review of each attack and its impact on the NDN functionalities. Furthermore, our goal is to give an up-to-date survey of recent NDN IDS with an in-depth analysis of their techniques and limits and to

identify open research issues for each attack, to give more opportunity for researchers to design better IDSs and secure the NDN architecture.

1.3. Our Contributions

Our main contributions in this paper can be summarized as follows:

1. A structured and comprehensive up-to-date review of the attacks that can target the NDN architecture with an analysis of the sub-categories of each attack and their impacts on the NDN structures.
2. A detailed survey of the detection and mitigation techniques recently proposed in the literature for the four classes of attacks with a new classification of the presented NDN IDS based on their attack detection algorithm.
3. An in-depth analysis of the limits of the presented NDN IDS and their comparison based on complexity, used topology, exposure to malicious nodes, etc.
4. Finally, we identify the open research issues related to each attack to assist future NDN security research directions.

1.4. Organization of the Survey

As shown in Figure 1, we have organized this survey hierarchically. Starting, in Section 1, by a complete overview of the NDN architecture: the essential components, the forwarding strategies, the naming schemes, content caching strategies, and we end up with a complete cover of the pre-existed security mechanisms in NDN. In Section 2, we did walk through most of the security threats for the NDN architecture, and we classified them in-depth by the security target component such as availability, confidentiality, and integrity. Then, we proceed to further studies of the process and the impact of each attack. In Section 3, we detail recent detection and mitigation mechanisms of each attack presented in Section 2; we classified state-of-the-art NDN IDS into four well-organized categories of the solutions studied in each category. A deep analysis of the challenges and limits of each NDN IDS is presented in Section 4. Finally, in Section 5, we elaborated on the open research issues related to NDN security.

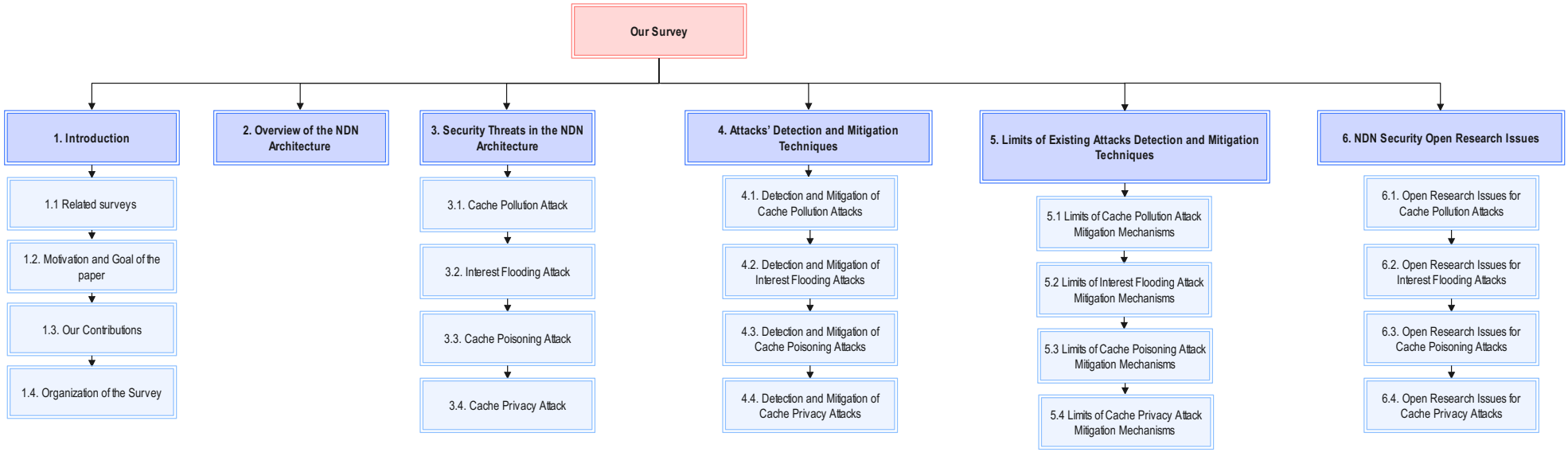

Figure 1. Survey structure.

2. Overview of the NDN Architecture

NDN is based on the content pulling model. In NDN, the node that requests a content is called *Consumer* and the node that provides it is called *Producer*. NDN relies on two type of messages, called the *Interest* packet and the content object or *Data* packet [20]. To manage data dissemination, each NDN node implements three main components:

- **Pending Interest Table (PIT):** It contains the interest packet entries and the corresponding list of incoming interfaces and "on hold" interests that have not yet been satisfied. Multiple incoming interfaces indicate that the same data are requested from multiple downstream users.
- **Content Store (CS):** The CS holds a copy of the Data packets that have been passed by the NDN router, and this copy will be used to satisfy subsequent interests that request the same content. This caching mechanism speeds up data delivery and decreases the network load.
- **Forwarding Information Base (FIB):** FIB entries associate prefix names to one or many output interfaces to which the interest packets should be forwarded.

As shown in Figure 2, a consumer requests data by sending an interest packet to the router. The router looks for such content in the Content Store (CS) if a cached copy exists; it sends it back to the consumer using the same interface through which the interest packet is received. If the content is not present in the CS, then the router checks the PIT entries, if an entry matches the interest prefix (the name of the requested content), the *Interest*'s incoming interface is added to this entry and the received *Interest* will not be forwarded, and it is going to be dropped [21,22]. If no entry has been found in the PIT, a new entry will be created, and the received *Interest* is forwarded based on the routing information of the FIB. Although, if no matching route is found in the FIB, a *Negative ACKnowledgments (NACK)* [23] with "No route" alert raises, the *Interest* will be dropped or becomes broadcast, based on the routing policy of the router [24].

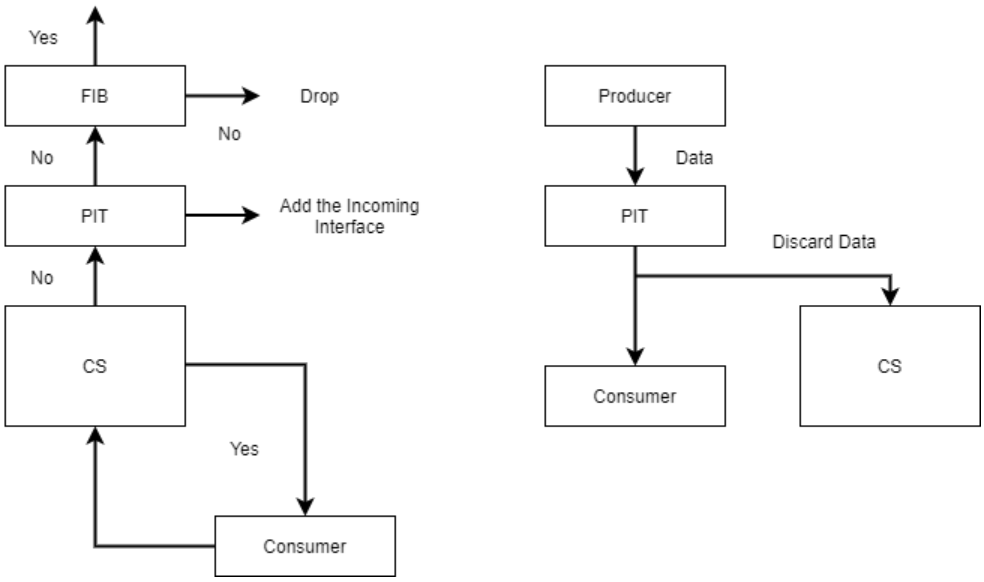

Figure 2. Packet processing.

Meanwhile, when a Data packet is received, the NDN router caches the content carried in the data packet and the related PIT entry is removed. Thus, finally, the data packet arrives at the requesting consumer using the reverse path of the interest packet that requests it.

Caching is one of the key features that makes the NDN architecture much more advantageous compared to the traditional TCP/IP architecture. Caching in NDN is different from the traditional web caching "Recency-based policies", which are based on the idea that the content has been demanded recently in a short period of time, in other words,

"short time locality". However, there is a high probability that the content will no longer be requested. NDN has different caching processes, policies, and metrics. NDN caching is much more optimized, and it manages to minimize both the bandwidth usage and the data retrieval delay, all along with congestion avoidance. To manipulate the caching in the NDN architecture, several caching strategies have been defined [25], such as:

- **Least Recently Used (LRU) (the content that has been less demanded is discarded):** this cache policy is popular due to its well-performed measures, and it increases the chance of the cache hits, where it stores the most recent data for a longer period of time.
- **Least Frequently Used (LFU):** second most used cache policy in NDN architecture, in which the cache decision is based on the content. LFU entails evicting the item with the fewest requests during the previous time window. Only the most frequent objects from that time period continue to stay in the cache.
- Other caching policies are also used, such as FIFO (First In First Out) and Random (decide randomly to cache or not a content).

Table 2, summarizes and compares the four presented caching strategies.

Table 2. Caching strategies in NDN.

Cache Policy	Cache Policy Categories	Cache Policy Usage
Least Recently Used (LRU)	Recency-Caching-Based	+++
Least Frequently Used (LFU)	Frequency-Caching-Based	++
First In First Out (FIFO)	Naive-Caching-Based	+
Random	Randomness-Caching-Based	+

Security in NDN is insured since the foundation of this architecture and each packet is transmitted while guaranteeing several security goals such as confidentiality and data integrity. In NDN, a Signature field is inserted in each Data packet and the content and the signature are binded together, to insure the verification of the data along the way of its transmission. When a data packet is received, the requesting consumer verifies the signature field. Although, the verification by intervening routers is not mandatory because the verification overhead of the signature might be high, and a router needs access to multiple public key certificates to trust the public key that verifies a content signature [26,27]. More precisely, the signature field of a data packet is composed of two sub-fields, namely *SignatureValue* and *SignatureInfo*, that are used to monitor authenticity and integrity. Any node that receives a data packet can utilize the information in the *SignatureInfo* to obtain the publisher's public key. This public key will then be used to decrypt the *SignatureValue*. After that, the result is compared to the hash of all other data fields. This procedure allows the consumer to check the data packet's integrity. Furthermore, in order to authenticate the content provider's legitimacy, the user must have credibility in the owner of the public key used to sign the data. The hierarchical name structure helps in the formation of trusting associations. For example, if we take a piece of content with the prefix */www/rnu/fsg/com/tn/index.html*, the signature then is made up by the owner of /www/rnu/fsg/com/tn domain, in which the key, is certified by the owner of /www/rnu/ domain as a result; it forms the chain of trust, which is linked by the KeyLocator [28], as shown in Figure 3. The available signature types pre-defined in NDN are:

- SHA-256;
- RSA signature over SHA-256;
- ECDSA signature over SHA-256;
- Hmac over Sha-256.

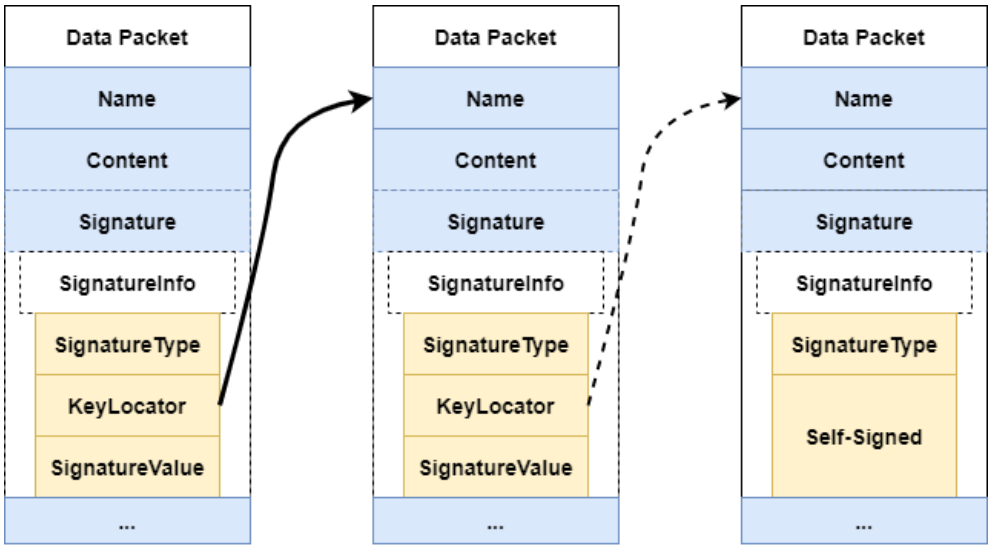

Figure 3. Chain of trust in NDN.

The NDN architecture's endurance can ideally be tested against the Internet's well-known Denial-of-Service (DoS) cyberattacks. Gasti et al. provide in [29] an in-depth study of NDN resistance to known attacks and TCP/IP vulnerabilities. Because NDN has no concept of addressing, it is resistant to attacks targeting network endpoints. As a result, common threats that overwhelm consumers' resources with traffic such as TCP SYN flooding attacks could not be performed in NDNs. Likewise, malicious payloads carried out by hacked routers that advertise erroneous routes such as prefix hijacking and Sub-prefix hijacking [30], could be easily identified and avoided thanks to the NDN's adaptive routing mechanism.

3. Security Threats in the NDN Architecture

Despite the high built-in security level of the NDN architecture, it still suffers from various attacks, as shown in Table 3; those attacks are well-defined in the following subsections:

Table 3. The effects of NDN attacks on the security goals.

Attack	The attacker entity	Target Security Goal	Target Entity
Cache Pollution	Consumer/producer	Availability	CS
Cache Privacy	Consumer	Confidentiality	CS
Cache Poisoning	Consumer/producer	Availability	CS
Interest Flooding	Consumer	Availability	PIT

3.1. Cache Pollution Attack

In a Cache Pollution Attack (CPA), the attacker tries to cache unpopular content in the Content Store (CS) in an attempt to make the cache unavailable to legitimate consumers. This attack mostly targets NDN routers Cache Hits. As shown in Figure 4, an *Attacker* node sends interest packets to change the priority of the content stored in the CS of nearby routers. This induces the caching of a large number of malicious packets in the CS of router R1 (malicious contents are represented in red in our example). This behaviour changes the priority of the content and increases the popularity of these malicious content items. As a result, this attack reduces the probability of obtaining legitimate content from the cache by legitimate consumers. This attack confuses the router from detecting such fake content, so it keeps them in the router, which results in *False Locality Attack (FLA)*, and that is because the attacker does not follow any specific pattern. Another type of this attack called the *locality disruption attack (LDA)* is based on several attackers' high frequency demanding malicious content. Moreover, CPA attacks can affect also the consumer side, by increasing the time to

obtain the desired content. This behaviour results in reduced bandwidth, increased data retrieval delay and in certain cases results in a Time-Out.

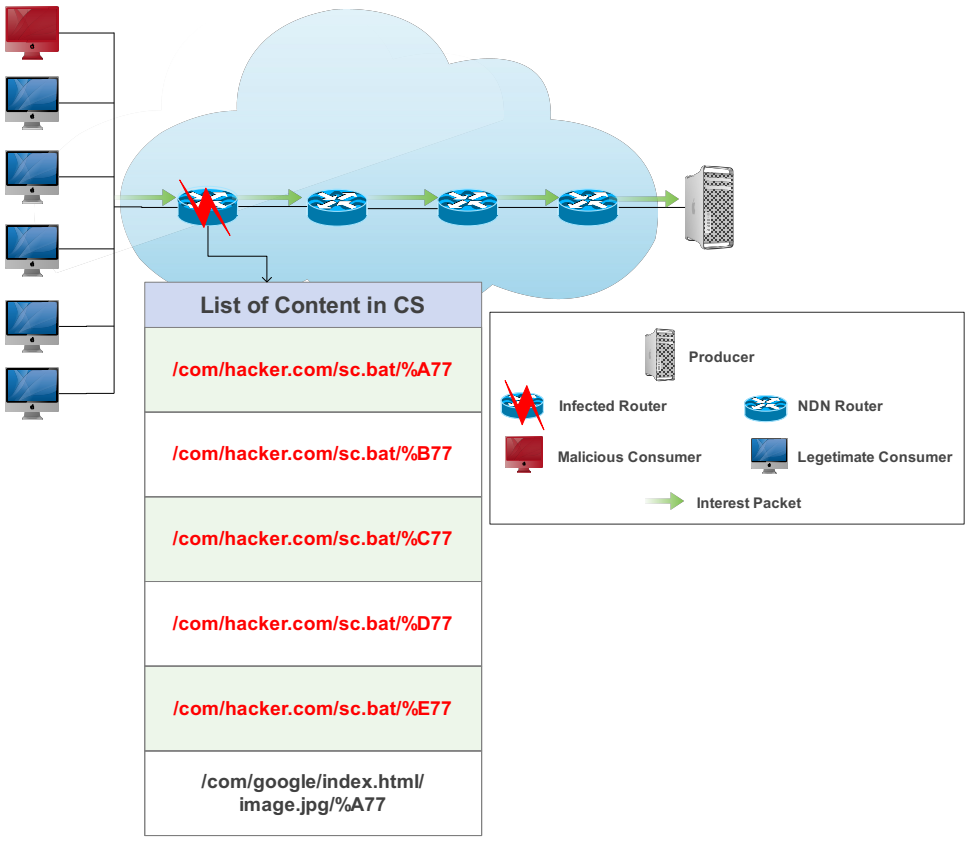

Figure 4. Cache pollution attack.

This attack is difficult to detect because it is hard to identify the attacker in the NDN architecture (NDN conserves the privacy of the consumers). In addition, the attacker does not follow a specific pattern, such as the amount of interests sent per second and the hierarchy of prefix naming. Moreover, the time of launching the attack is not stable and, in several cases, follows different strike timing. Deng et al. [31] classified CPA into two main categories: *Locality Disruption Attack (LDA)* and *False Locality Attack (FLA)*. For LDA, the malicious consumer requests masses of junk content. This type of content gets cached in the CS and this malicious consumer keeps requesting it to keep it in the CS. In the case of FLA, the malicious node requests already existing content with less popularity in order to prevent other legitimate content from getting higher popularity and falsify the priority rules implemented in the caching policies, which forces it to be useless.

As explained in [10], CPA essentially affects the *Cache Hit Ratio (CHR)* by decreasing the hit ratio of legitimate content. Ref. [8] explains that in several cases, the CHR of legitimate requests could decrease to 0% in edge routers, i.e., routers that are directly connected to the attackers. CPA attacks also affect the *Average Retrieval Delay (ARD)*, which greatly increases compared to the normal state. This leads to a direct effect on the legitimate consumer and may result in an unnecessary re-transmission of the legitimate interest.

Other evaluation metrics have been suggested in our previous work [32], such as the *Hit Damage Ratio (HDR)*, which determines the efficacy of the CPA attack on the caching core of NDN by measuring the ratio of normal users' hit ratios in the absence of the attack by hit ratios in the presence of the attack; if the value is close to 1, the attack is more highly effective, and if it is close to 0, the attack has less effect on that component. The results

of HDR were measured in both the LDA and FLA cases. They show a huge impact on the caching process of legitimate content and the critical impact of CPA on confusing the caching strategies in the CS.

3.2. Interest Flooding Attack

As shown in Figure 5, an Interest Flooding Attack is a type of attack that targets the Pending Interest Table (PIT), and aims to send a huge amount of interests into the desired router, which forces all these routers to create entries in their PIT that remain open during the attack. As a result, the PIT entries will no longer be available for legitimate consumers and keep dropping each packet sent by them [10,33].

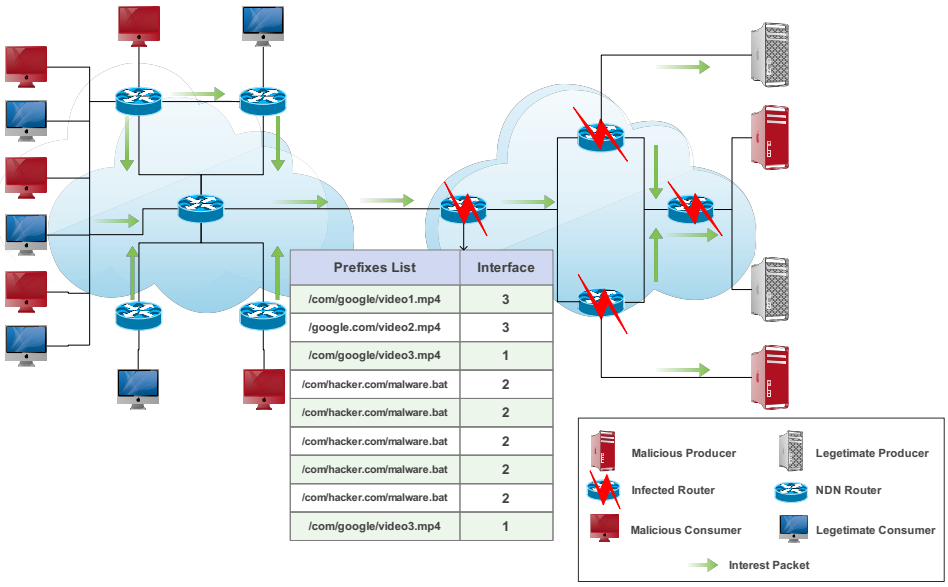

Figure 5. Interest flooding attack.

Three types of such attacks have been specified: existing or static, dynamically-generated and non-existing attacks. For the first type, the attacker sends an interest of an existing content item that will be cached in the CS and this will open a small number of entries in the PIT. The second type is based on dynamically generating different interests with high frequency. This type of attack is more efficient than type 1. For type 3, the malicious node sends interest packets of non-existing content. This type ensures that the router creates a higher number of PIT entries that remain open until the time-out. This type is more severe than types 1 and 2 because PIT entries remain open for a longer duration.

Two kinds of IFA have been defined in [34], namely (1) Basic IFA (BIFA) and (2) Collusive IFA (CIFA) (see Figure 6). As for BIFA, the attack is targeting the PIT table, where its main goal is to saturate the PIT entries and force it to be unavailable to satisfy the next requests. In the second hand, CIFA aims essentially to fill the intermediate routers with legitimate content from the malicious producer. The main difference is that the attack is more likely undetectable and none recognizable by the attack detectors, which makes it have a higher impact compared to the basic one. Recently, Zhijun Wu et al. [35] have extended the work to the new improved technique by combining the BIFA and CIFA to release I-CIFA.

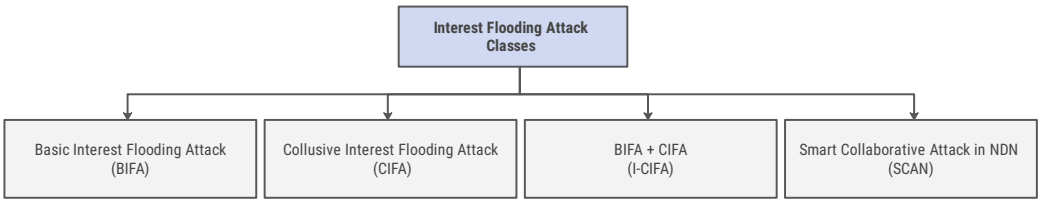

Figure 6. Interest flooding attack classes.

Madhurima Buragohain et al. [36] introduced a new category of attack called SCAN (Smart Collaborative Attack in NDN architecture). This attack mainly aims to affect the QoS of the legitimate consumers in the NDN network. The SCAN system model consists of two main phases:

1. **Pre-Attack phase:** consisted of four main essential metrics: Minimum Retransmission Wait Time (MRWT), Minimum Interest Frequency (MIF), the Minimum number of pieces of content stored in malicious producers and the topology characteristics.
2. **Main Attack phase:** consisted of two steps such as: collecting the number of set of prefixes stored in the malicious producers and then the malicious consumers dynamically set the interest frequency, so it cannot be obviously detected, and that is managed by setting a pre-defined threshold.

3.3. Cache Poisoning Attack

As shown in Figure 7, Cache Poisoning Attack mainly aims to inject either fake or corrupted content into the router, which remains in the routers and keeps spreading into the neighbour nodes. Processing a content item in a line speed timing, leads the NDN router to be unable to deeply verify the malicious content [33]. This attack can be performed either by a compromised router, which spreads poisoned content in reply to the interest packets. The other neighbour routers will cache this content, which will be accessed later by other consumers. The attack is highly dangerous, as it can distribute poisoned content through compromised publishers and routers, and spread fake or corrupted content very quickly. The main security concern in a Cache Poisoning Attack is the availability of the content.

3.4. Cache Privacy Attack

In a Cache Privacy Attack, the attacker tries to access the cached content present in the CS to find out whether this content has recently been accessed by certain legitimate consumers or not. When the attacker knows the time of the access of such content, the attacker associates this content either to malicious consumers or to different legitimate consumers. This attack breaks the privacy of legitimate users, and it requires several steps to apply it, starting from the "Enumeration" of the content present in the CS and other information such as the access time of each content item in the CS cache, then requesting such content by either predicting the next pattern of interest or requesting the same content.

The authors of [37] studied the cache privacy attack for CCN. As shown in Figure 8 they identified three types of cache privacy attack, such as a Timing-Based Attack (TBA), Object Discovery Attack (ODA) and Data Flow Cloning Attack (DFCA).

Moreover, two new type of attacks have been identified recently in the cache privacy attack category, namely (1) Cache Side Channel Attack (CSCA) and (2) NDN Traffic Analysis Attack (NTA), where both of these attacks can be performed simply to the NDN router to identify the content existing in the cache of the CS.

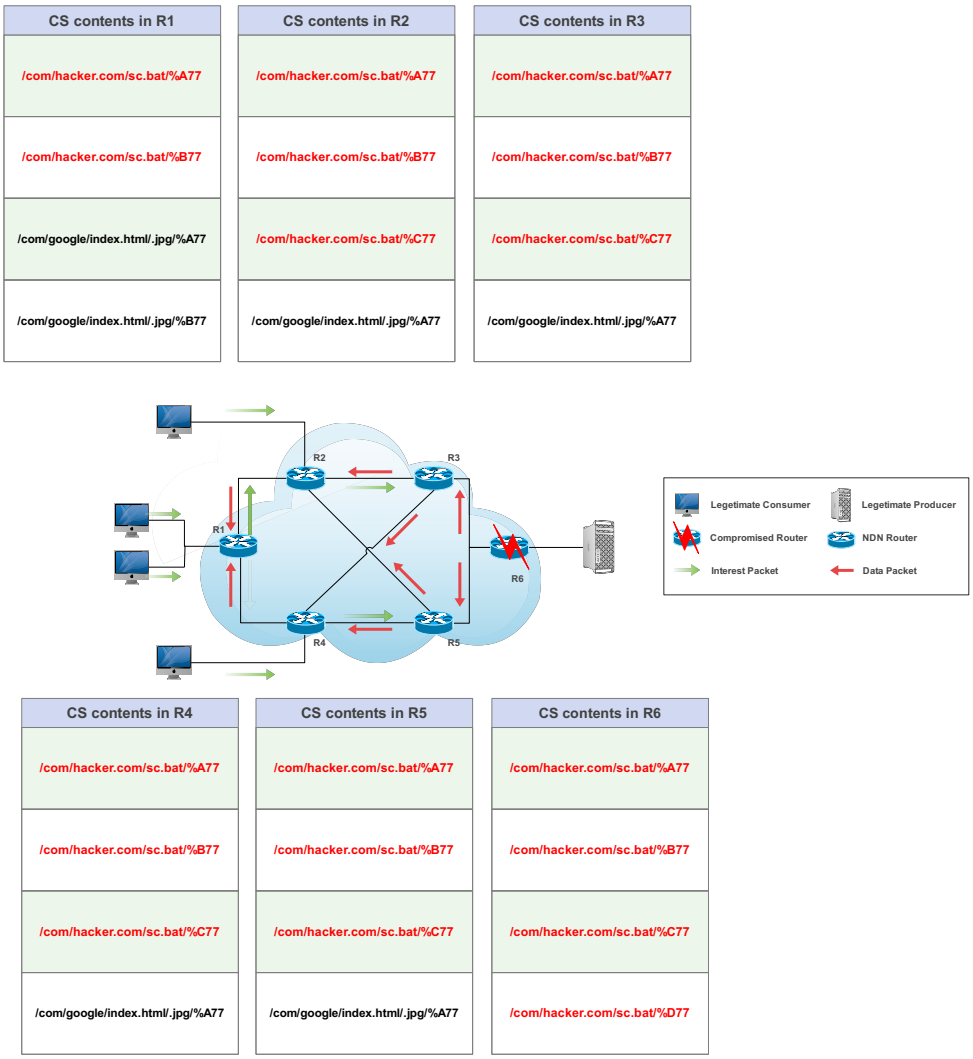

Figure 7. Cache poisoning attack.

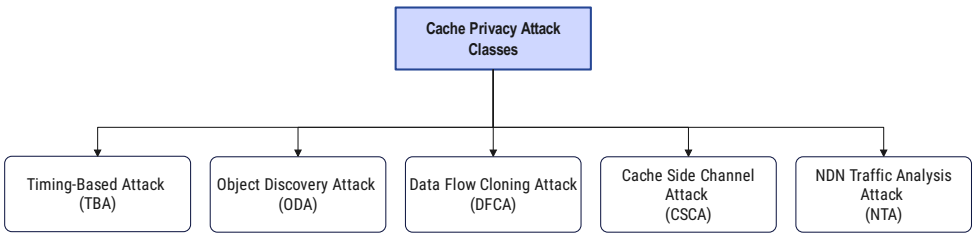

Figure 8. Cache privacy attack classes.

3.4.1. Timing-Based Attack (TBA)

In TBA, the attacker tries to enumerate the content presented in the CS. The attacker finds the hit time of the cache by requesting the same content twice. The first request caches the content. The second request is satisfied from the cache, as the content has already been cached. The attacker requests the desired content to check whether it remains in the cache or not. If it is cached and the cache hit occurs by legitimate users, the attacker can interpret that the consumer who requested this content is linked to such content.

3.4.2. Object Discovery Attack (ODA)

In ODA, the attacker sends an interest that has the root namespace "/"; then, the router responds with a random content from the cache of the nearby router or any other routers by specifying the HopLimit. After that, it constructs the prefix based on the recent result of the enumeration. Taking the example of an attacker who sends an interest with root namespace "/", the CS cache responds with "/com/website/www/media/video.mp4/%77"; then, the attacker changes the prefix into "/com/website/www/media/" to obtain a list of content in that namespace, and so on.

3.4.3. Data Flow Cloning Attack (DFCA)

In DFCA, the attacker tries to use ODA to enumerate the cache content. He targets the ongoing flow interests and predicts the next namespace that the legitimate consumer still has not requested and sends it to change the ongoing flow interest on his side. This attack is applied, for example, in Voice-over-CCN applications. Globally, a Cache Privacy Attack targets the confidentiality of content in the CS.

3.4.4. Cache Side Channel Attack (CSCA)

The time elapsed between sending the interest packet and getting the data packet is known as the NDN-RTT (NDN round trip time). In the normal NDN forwarding process, when the consumer sends an interest requesting a desired content, it verifies first the Cache of the CS if that content existed in the CS Cache, then, it sends it back to the consumer who requested it in the NDN-RTT pre-defined time.

The term Cache Side Channel Attack (CSCA) refers to the method that the attacker can use these RTT discrepancies to determine which content has been or has not been added to the CS.

More explicitly, CSCA is an attack technique wherein the attacker monitors user behaviour by comparing the access times for cache hits and misses using this NDN-RTT, then uses that information to deduce the sensitive data of the users.

Ertugrul Dogruluk et al. [38] studied the impact of the CSCA by calculating the NDN-RTT using the TimeOut Impact Value. An NDN-RTT measurement starts at each receiver of an interest in the NDN router. A counter starts counting until the receipt of the desired data packet.

3.4.5. NDN Traffic Analysis Attack (NTA)

Alberto Compagno et al. [39], studied a new type of attack called NDN Traffic Analysis that falls in the category of "Proactive Cache Privacy Attack"; this attack stands on three essential steps:

- **Preparation:** Consisted of the necessary parameters to push the malicious content into the 1-Hop Router.
- **Content Loading:** Force the 1-Hop NDN router to cache the malicious content.
- **Traffic Analysis:** Monitor which content has been requested from the CS cache.

4. Attacks' Detection and Mitigation Techniques

4.1. Detection and Mitigation of Cache Pollution Attacks

As shown in Table 4, the first early mechanism to detect CPA has been proposed by Mengjun Xie et al. in [40] and is called, CacheShield. The main process of this mechanism is based on a shield function which calculates the frequency of receiving a content item and indexing it with each associated content based on its popularity and its naming-prefix, then it verifies if such content is already cashed. If so, it delivers it to the consumer that requests it; otherwise, it stores a slice of this content and resets its frequency counter. By respecting the chronology, Guo et al. [41] suggested a mechanism based on path diversity. The goal of this mechanism is to base its decision on the collected information related to each data

packet stored in the cache of the CS. The collected parameters consisted of the hit time of a content item *o.hit*, the path traversed of a content item *o.PathTracker*, the damage ratios on backbone router, false positive of the decision and false negative error ratios of the decision as illustrated in Figure 9.

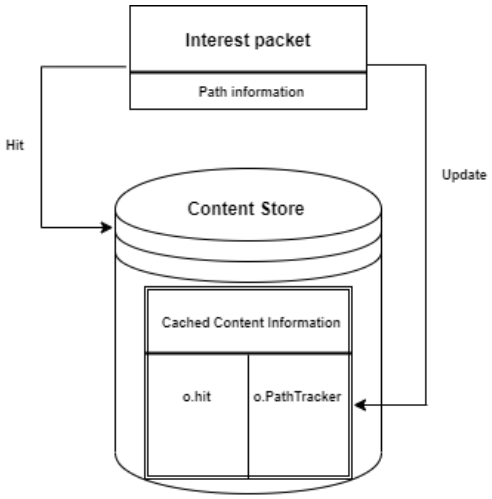

Figure 9. Path traversal mechanism.

Then, this mechanism compares the normal state before the attack and the attack state using a "PathTracker". The authors defined a threshold, which is compared with the value obtained. If the value is higher than the predefined threshold, the attack is detected. Kamimoto et al. [42] suggest a different mechanism based on prefix hierarchy and called the *Cache Protection Method based on prefix Hierarchy (CPMH)*. This mechanism goes through three main steps: extracting the malicious nodes' prefixes and saving them on a BlackList. This is conducted by calculating the Request rate Variation for each prefix (RVP), and to avoid obtaining a false positive, the mechanism also calculates the Weighted Request rate Variation for each prefix (WRVP). The next step is to eliminate the prefixes existing on the BlackList. Finally, each interest that requests a blacklisted prefix will be dropped. Zhang et al. [43] propose a mechanism based on the Coefficient of Variation (CV), where the decision is made based on several parameters, namely: The prefix received x, the interface from which the interest is received and the frequency of sending such an interest. These collected statistics are used to calculate the $CV_i(x)$ of an interest i with prefix x; if the value is low, it is more likely that it is from an attacker source, if not, it is suitable to be cached. The decision in this mechanism is made on the data, which means either to cache it or not to cache it. Another CPA detection mechanism has been proposed in [44] and it is based on clustering. Globally, this mechanism collects some information upon the running of the mechanism, for-instance, the number of interests received, the number of interests for the same content and the time interval between two consecutive requests for the same content, then it classifies the interests into two different clusters based on the probability of an interest i appearing and the AVG time interval between two interests requesting the same content. This helps in the final step to determine the type of the attack (LDA or FLA); then, each router broadcasts to its neighbours a list of suspicious content items in order to avoid them being cached.

The authors of [45] suggested a detection mechanism called ICAN (Intrusion detection system for CPA attack in NDN architecture) based on the metrics of performance including the average cache hit ratio, average interest inter-arrival time, hop count and prefix variation that basically stands by monitoring dynamically the variation of those metrics to decide the appearance of the attack in different realistic network topologies. This solution demonstrated a high efficiency compared to previous mentioned solutions in terms of conserving router resources, the conservation of the user's privacy and the high accuracy of detecting the attack. Another detection approach is Based on Probability Decision (BPD).

This approach includes a mechanism called the Randomness Check [46]. This solution starts by creating a matrix composed of the names and how many times each content item is received. Upon receiving some content, it increments the value of that content in the matrix. If the requested content goes higher than the predefined threshold, it is suspected of being an attack. In the case of an attack, the content is eliminated, otherwise, it stays for future demand.

A detection mechanism called Adaptive Neuro-Fuzzy Inference System (ANFIS) has been proposed by Karami et al. [47], which mainly aims to change the caching replacement policy. The main step in this mechanism starts by collecting data related to each interest, which lead to feeding the features of its neurones that are constituted by five fuzzy layers; as result, a goodness value is collected on each interest. This value is taken into consideration to decide whether to cache the content or remove it from the cache of the CS.

Kumar et al. [48] proposed a mechanism to detect CPA called the Interface-Based Popularity Caching (IBPC), which goes through the collection data based on the number of interfaces that receive a content item in a certain period of time. IBPC focuses on calculating the number of users requesting a content item using the Exponentially Weighted Moving Average (EWMA) to define the popularity of the content over a given period of time, considering that the number of attackers is smaller than the number of legitimate consumers. The new technique of detecting CPA has been introduced by Lei et al. [49], where this mechanism relies on collecting information such as:

- Data validator;
- Provider CS;
- Content name;
- Task prefix;
- Digest prefix.

By collecting this information, the mechanism constructs blocks on each content item and verifies it by comparing it with a predefined threshold to either proceed in caching the content or deny it from being accessed to the cache of the CS.

Akanksha Gupta et al. [50], proposed a technique to detect CPA by utilizing the K-Means Clustering technique in their method. The authors use the information related to each interest, and classify it as either LDA or FLA. An attack table is implemented, where it contains each suspicious interest. In order to mitigate the attack, the previously mentioned attack table is transmitted to the neighbour nodes The data packet in the associated interest packets will not be cached by those NDN routers. Cache nFace is a mechanism suggested by Andre Nasserala et al. [51]. The main goal of this mechanism is to divide the CS cache into multiple sub-caches and assign each sub-cache to a different NDN router interface. Similarly, as suggested in [51], Jie Zhou et al. [52] proposed a mechanism based on the Cache Partition. The authors divide the CS cache into two parts and associate each content with a popularity rate.

Network commencement, interest packet publication, data packet publication, and data packet verification are the four phases that make up the process of advanced hierarchical identity-based security mechanisms by blockchain (AHISM-B) [53]. The mechanism globally poses its foundation based on the hierarchical identity-based cryptography (HIBC) in combination with the blockchain processes.

A classification of the different CPA detection and mitigation techniques is shown in Figure 10.

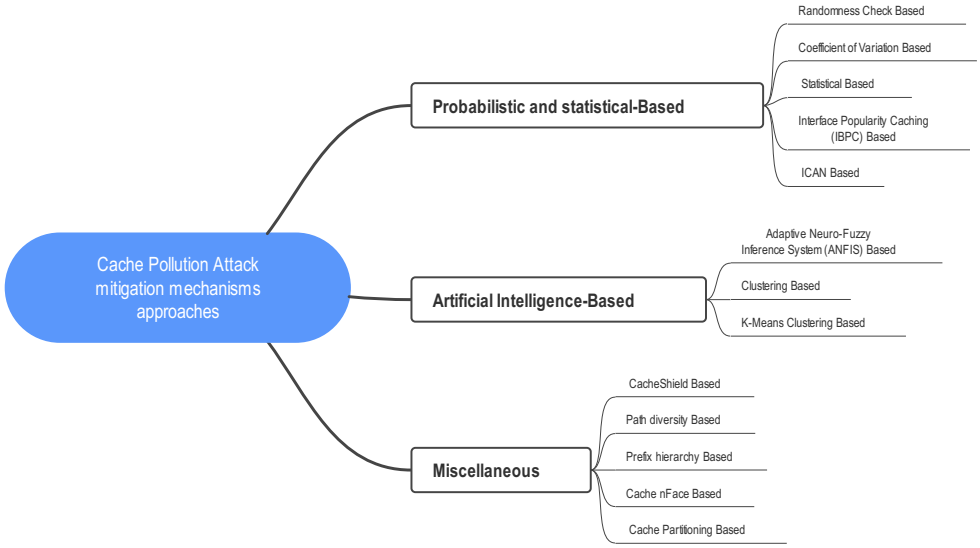

Figure 10. Cache pollution attack approaches.

Table 4. Comparison of CPA detection and mitigation mechanisms.

Paper Reference	Year of Release	Method
Mengjun Xie et al. [40]	2012	CacheShield-Based
Park et al. [46]	2012	Randomness Check-Based
Karami et al. [47]	2015	Adaptive Neuro-Fuzzy Inference System (ANFIS)-Based
Guo et al. [41]	2016	Path diversity-Based
Kamimoto et al. [42]	2016	Prefix hierarchy-Based
Zhang et al. [43]	2017	Coefficient of Variation-Based
Andre Nasserala et al. [51]	2019	Cache nFace-Based
Yao et al. [44]	2020	Clustering-Based
Lei et al. [49]	2020	Statistical-Based
Jie Zhou et al. [52]	2020	Cache Partitioning-Based
Akanksha Gupta et al. [50]	2021	K-Means Clustering-Based
Kumar et al. [48]	2021	Interface Popularity Caching (IBPC)-Based
Hidouri et al. [45]	2022	ICAN-Based
Bing Li et al. [53]	2022	Blockchain-Based

4.2. Detection and Mitigation of Interest Flooding Attacks

In IFA and as shown in Figure 11, some mitigation mechanisms have been suggested. In [54], the authors propose using PIT entries as the main parameter when it goes beyond a predefined threshold. Then, the NDN router lists the unwanted interest packets with the longest prefix and sends the associated data packets to the backbone routers, which process the interface where this interest comes from. Compagno et al. [55] proposed a mechanism to detect IFA by utilizing the ratio of interests that come from an interface i, as the ratio of interests that go out from an interface i^* and PIT capacity for each interface i. The mechanism calculates the goodness value using those parameters, if it goes more than a pre-defined threshold, an attack is detected so the NDN router revokes such an interest and a notification message is sent to the neighbour router about the malicious interest. A neural network mechanism has been suggested by [56], and the features used in detection are:

- The number of arriving data packets;

- The number of arriving interest packets;
- The number of outgoing data packets;
- The number of outgoing interest packets;
- The number of satisfied interest packets;
- The size of the PIT entries.

This mechanism has been evaluated on the DFN topology and has demonstrated a high accuracy compared to the recent, previously mentioned mitigation mechanisms.

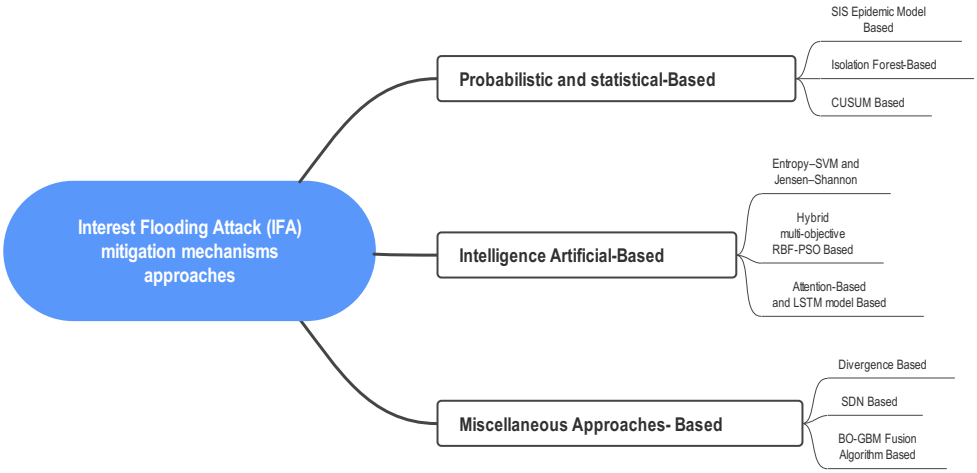

Figure 11. Interest Flooding Attack approaches.

The authors in [34] tried to step counter this attack by suggesting a method relying on the Cumulative Sum (CUSUM) algorithm. This detection algorithm relies on monitoring the time for the change of the pattern of the timing-requested content. As the authors claimed, in a normal state, the time it takes for a producer to receive a request is absolutely inferior to the lifetime of that interest, while under an attack state, the malicious producer will try to match the same lifetime received by the malicious consumer, and as a result it will send that data associated with the requested interest in a longer time period compared to the normal state. CUSUM also poses on the variation of the benchmark of the interest, where those small changes cannot be detected easily.

Kai Wang et al. [23] introduced a new technique of detecting the presence of malicious interests in the network by applying the Heterogeneous N-intertwined Mean-Field Approximation of SIS epidemics on the NDN Networks model (HNIMFA). This model is going through employing the in-homogeneous susceptible-infected-susceptible (SIS) process, in order to model the malicious transmissions of the interest packets between the infected router and both the malicious consumer and the malicious producer.

In other hand, Ting Zhi et al. [57] proposed a mechanism based essentially on the Entropy-Support Vector Machine (SVM) and Jensen-Shannon (JS) Divergence to counter the IFA attack in the NDN network. As shown in Figure 12, their technique goes through three main essential steps.

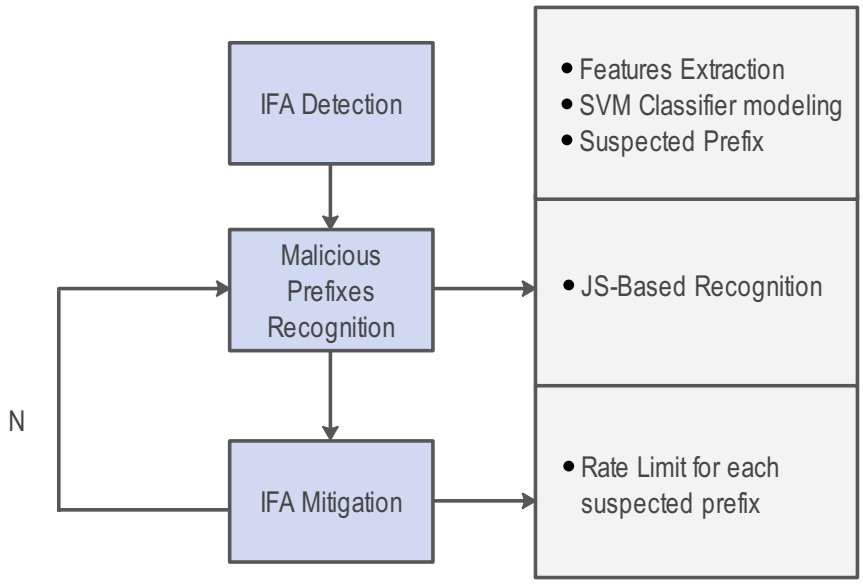

Figure 12. SVM Entropy and JS divergence based for detecting IFA.

A. Benmoussa et al. [58], introduced a mitigation mechanism called MSIDN (Mitigation of Sophisticated Interest flooding-based DDoS attack in NDN). This mechanism is composed of five main steps: Control interest packet, Hop-by-Hop signing and verification, Producer-based IFA mitigation, Router-Based IFA mitigation, Blocking malicious nodes.

Choose to kill the interest flooding attack (ChoKIFA) is a mechanism that has been suggested by A. Benarfa et al. [59], where ChoKIFA's main goal is to locate and drop the malicious interests by taking use of the PIT state, which offers sufficient statistics on interest packets, as the ones that are coming in and going out. As soon as an interest reaches a router, ChoKIFA selects one random interest from the PIT and compares it with the one that has just arrived. Both interests are dropped if they are a part of the same traffic flows. If not, the randomly selected interest is kept in the PIT and the incoming one is stored there with a probability based on the PIT occupancy level.

M. Alhisnawi and M. Ahmadi [60] introduced a new mechanism for detecting and prevention the presence of IFA based on the use of the Software Defined Networking architecture (SDN). This mechanism consists of five main components: Content Provider, NDN Controller, Content Provider Router, Entering and departing NDN router, Intermediate NDN router (InR) and Edge NDN router (EdR).

This detection mechanism is utilizing most of the previous mentioned routers to collect two main metrics such as: PIT expiration rate (PER) and PIT occupancy rate (POR).

A further investigation to realize a good IDS counter IFA has been made by Zhijun Wu et al. [61], which relies on two main parts:

1. **Detection strategy:** An initial stage of detecting the attack is to launch the network functionality in a normal state in order to store the history of the traffic transmission. The next stage compares the historical traffic transmission with current ones, and those metrics are as follows: The time window rolls, the throughput and the existent time of PIT entry in the current time window.
2. **Mitigation strategy:** In case an anomaly is suspected, the mechanism starts to measure the rate of the PIT allocated. If this value is superior to a pre-defined threshold, the PIT entries that remained for a longer time will be reset.

A prediction error to detect the presence of CIFA has been suggested by Liang Liu et al. [62], whereupon, when launching the network simulation, the mechanism starts to collect information from PIT component and a comparative analysis defines which interests need to be discarded and refresh the interface allocations.

Guanglin Xing et al. [63], suggested a mechanism based on the Isolation Forest (IForest). The detection process for this mechanism is composed by isolation trees (binary tree structure). A number of prefixes are chosen to construct a dataset to utilize for the training. The dataset is labelled, as follows:

- Number of sent interest packet.
- Number of received data packet.
- Number of entries recorded in the PIT.
- Number of expired PIT entries in the PIT.

Shorter ITrees need to be made in order to optimize the findings; in this stage, several results can be conducted:

- If the score value of the data prefix is closer to 1, the prefix is suspected.
- If the score value is under 0.5; the prefix is valid.
- If all prefixes have the score value equal to 0.5, no attack exists in the network.

The mitigation process is managed by sending a notification to all downstream nodes of the malicious interest. This will trigger the restriction of the transmission of a certain malicious prefix that has been suspected.

Xin Zhang et al. [64], used the Attention-Based technique all along with Long Short Term-Memory (LSTM), to retrieve the IFA attack behaviour on the network.

Another mechanism has been extended by Zhijun Wu et al. [65], to detect the presence of I-CIFA. The proposed technique is based on the Bayesian optimized - Gradient Boosting Machines (BO-GBM) Fusion Algorithm. A logic Regression is used to make the resulted dataset stacked as the training set of the next-layer model through stacking method.

Table 5 summarizes most of the types of IFA mitigation mechanisms, the motivations and the main methods.

Table 5. Comparison of IFA detection and mitigation mechanisms.

Paper Reference	Year of release	Type of Attack	Method
Dai et al. [54]	2013	BIFA	Probabilistic-Based
Compagno et al. [55]	2013	BIFA	Probabilistic-Based
Karami et al. [47]	2015	BIFA	Neural Network-Based
Kai Wang et al. [23]	2020	BIFA and CIFA	SIS Epidemic Model-Based
Ting Zhi et al. [57]	2020	BIFA	Entropy–SVM and Jensen–Shannon Divergence-Based
Ahmed Benmoussa et al. [58]	2020	BIFA	Statistical-Based
Mahmood Ahmadi [60]	2020	BIFA	Probabilistic-Based
Zhijun Wu et al. [61]	2020	BIFA and CIFA	Statistical-Based
Liang Liu et al. [62]	2020	CIFA	Prediction Error
Abdelmadjid Benarfa et al. [59]	2021	BIFA	SDN-Based
Guanglin Xing et al. [63]	2021	BIFA	Isolation Forest-Based
Karami et al. [56]	2022	BIFA	Hybrid multi-objective RBF-PSO-Based
Al-Share et al. [34]	2022	BIFA, SMART and CIFA	CUSUM-Based
Xin Zhang et al. [64]	2022	BIFA and SMART	Attention-Based LSTM model-Based
Zhijun Wu et al. [65]	2022	I-CIFA	BO-GBM Fusion Algorithm-Based

* Basic IFA is labeled **BIFA**.

4.3. Detection and Mitigation of Cache Poisoning Attacks

Figure 13 explains in depth the Cache Poisoning Attack approaches, where the detection of the Cache Poisoning Attack is hard and expensive in terms of the resources of the NDN router, which led Gasti et al. [29] to propose a mechanism based on Self-certifying the data and the interest packets. Certain information is collected, including the hash of the content associated with the mechanism of each passing data, the Content Name and the signature of the content. This makes it possible to compare both the data collected of the interest and the data packet. Kim et al. [66] extended the work by reducing the overload on the NDN routers caused by these extensive verifications. The mechanism mainly aims to verify only the content existing in the cache: in the case of a cache hit, the verification is applied, otherwise, it is not. Ghali et al. [67] investigated the attack by associating each content item in the cache with a rank; this rank is variated by the number of requests for each content The lower the value, the higher will be the attack probability.

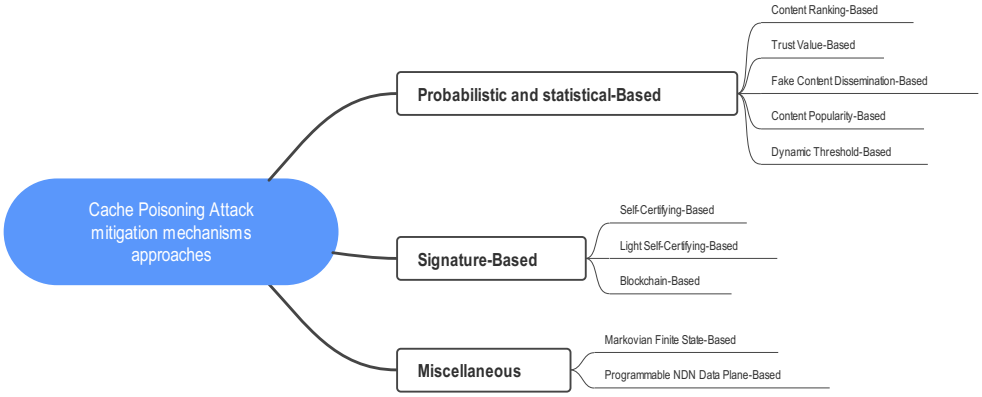

Figure 13. Cache Poisoning Attack approaches.

As specified in Table 6, Zeinab Rezaeifar et al. [68] introduced a new technique to detect the presence of the Cache Poisoning Attack using a Trust-based method. The mechanism is essentially based on three main metrics: the popularity of the content message, negative feedback, and the credibility of peers. The router determines whether to reply to or reject an Interest message depending on the reputation of the peer who sent it after defining the credibility of each peer. Additionally, the suggested approach stands on a new added field named the ”Exclude field“, which includes one or more hashes of undesirable elements in the interest messages as negative feedback. Finally, depending on the trusted model, which incorporates these three variables, the routers choose whether to cache or discard the requested content.

Table 6. Comparison of cache poisoning attack detection and mitigation mechanisms.

Paper Reference	Year of Release	Affecting Component	Method
Gasti et al. [29]	2013	MConsumer	Self-certifying-Based
Ghali et al. [67]	2014	MConsumer	Content Ranking-Based
Kim et al. [66]	2015	MConsumer + MRouter	Light Self-certifying-Based
Zeinab Rezaeifar et al. [68]	2018	MConsumer	Trust value-Based
Jerzy Konorski [69]	2019	MConsumer	Markovian Finite State-Based
Barun Kumar Saha et al. [70]	2020	MConsumer	Fake Content Dissemination-Based
Stanislaw Baranski et al. [71]	2020	MConsumer + MRouter	Blockchain-Based
Vishwa Pratap Singh et al. [72]	2021	MConsumer	Trust Value-Based
Saddam Hussain et al. [73]	2021	MConsumer	Self Certification-Based
Min Feng et al. [74]	2021	MConsumer	Content Popularity-Based
Adnan Mahmood Qureshi et al. [75]	2021	MConsumer	Dynamic Threshold-Based
Hou et al. [76]	2022	MConsumer	Programmable NDN Data Plane-Based

* Malicious consumer and malicious router are labelled **MConsumer** and **MRouter**.

In [70], Barun Kumar Saha et al. suggested a Fake Content Dissemination Based mitigation technique. To detect Cache Poisoning Attacks, the proposed detection module uses several metrics such as:

- Interests satisfaction ratio.
- Average latency.
- Data receiving ratio.
- CS size capacity.

However, the detection module is inspired by pre-existing solutions; the main contribution of the authors is the proposed mitigation mechanism that relies on a dynamic BlackList (BL). In fact, instead of relying on the feedback of one single node to store the suspected prefix name in the BL, each node communicates with its neighbors to record any detected suspicious prefixes in their blacklists.

Jerzy Konorski, suggested in [69] a method to mitigate the Cache Poisoning Attack by combining two main protocols, proposed as follows:

- **External Infection Protocol (EIP):** This protocol is used to grant access for the External Infection of the network nodes one-by-one; this protocol defines the trust value of each node in the network.
- **Internal Infection Protocol (IIP):** This protocol is used to launch the internal infection schema calculation based on Markovian finite state.

Further investigation has been made by Stanislaw Baranski et al. [71], where they utilized the BlockChain-Based mechanism to mitigate the NDN routers from Fake Data poisoning. In order to make each legitimate content valid in the next transaction, the Proof-of-Time ensures that each content is labelled with four main fields in each block:

- **Content Hash:** made in order to authenticate;
- **Producer Public-key**;
- **Previously Block hash**;
- **Signature:** contains the signature of the early mentioned content credentials.

A Fuzzy-Reputation Trust model has been proposed in [72] by Vishwa Pratap Singh et al. to find out the suspected content. The authors proposed a trust value calculated on each

incoming content and the previous content. The obtained trust value increase or decrease the reputation of the content $Repu_i$. In this step, three main scenarios can be adopted:

- **First scenario:** if $Repu_i > \beta$ then the content is safe;
- **Second scenario:** if $Repu_i < \alpha$ then the content is infected;
- **Third scenario:** if $\alpha < Repu_i < \beta$ then a fuzzy approach needs to be used;

where α and β are two pre-defined thresholds.

Saddam Hussain et al. [73] proposed a novel mechanism to prevent Cache Poisoning Attacks by adding a different authentication mechanism that is based on self certification in addition to the main signature. In this approach, each content adopted an ECC (Elliptic Curve Cypher) and the validation needs to be made in each path NDN router.

Min Feng et al. [74] suggested a mechanism to mitigate Cache Poisoning Attacks based on content popularity. The authors defined five popularity levels: Most Popular, Popular, General Popular, Not Popular and Least Popular. The mechanism decides to cache a content based on the obtained popularity level. Two main metrics are used to measure the popularity of the content, namely: "Content popularity level on the router" and "Content popularity in other neighbour routers cache".

Adnan Mahmood Qureshi et al. [75] defined a method to detect the source of anomaly on the NDN routers based on dynamic threshold values. The authors created a queue where instead of relaying using the standard forwarding mechanism, they verify each interest using a pre-defined threshold.

In [76], the authors classify the consumers based on the programmable NDN data plane. Two classes of consumers are defined:

- **VIP-Consumer:** one who demands the content urgently.
- **NonVip-Consumer:** normal basic consumer.

The metrics used to classify the consumers are the number of interests requested for an important content that is labelled as Assured Cache (AC) and the average local content popularity difference. If it is a NonVip-Consumer who demands a content, it will be labelled as No Cache (NC); otherwise, if the popularity of NonVip-consumer is high, it will be considered as a VIP-consumer.

4.4. Detection and Mitigation of Cache Privacy Attacks

Detection and Mitigation of Timing-Based Attack (TBA)

As shown in Figure 14 where in case of Cache Privacy Attacks , [77] proposed a mechanism relying on disabling the *scope* and the *exclude* because the malicious node is able to perform the attack by utilizing those two fields. More precisely, the *exclude* field guarantees that the attacker obtains the desired content without obtaining the same content covered from the first attempt, and the *scope* field allows the attacker to request the root namespace, which makes it easy for him to locate the desired content.

Ntuli et al. [78] suggested checking the interest and the content associated to the same prefix in order to identify the attack probability. The attack is suspected by the increase in the cache hit and the frequency of sending an interest, so it denies such an interest from accessing the content. [79] extended the work presented in [78] by adding other parameters, such as repeated requests for multiple content in a short period of time. A predefined threshold is used: if the goodness value decreases, an attack is suspected. Kumar et al. [80] proposed to decide either to add a delay on the malicious interest or to let it pass by judging through a static defined prefix hierarchy. This mechanism is applied for a Timing-Based Attack (TBA).

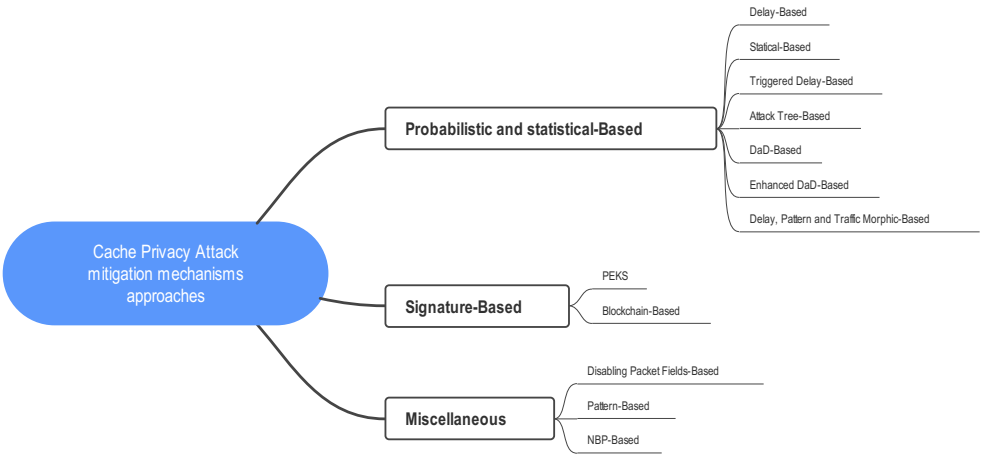

Figure 14. Cache Privacy Attack approaches.

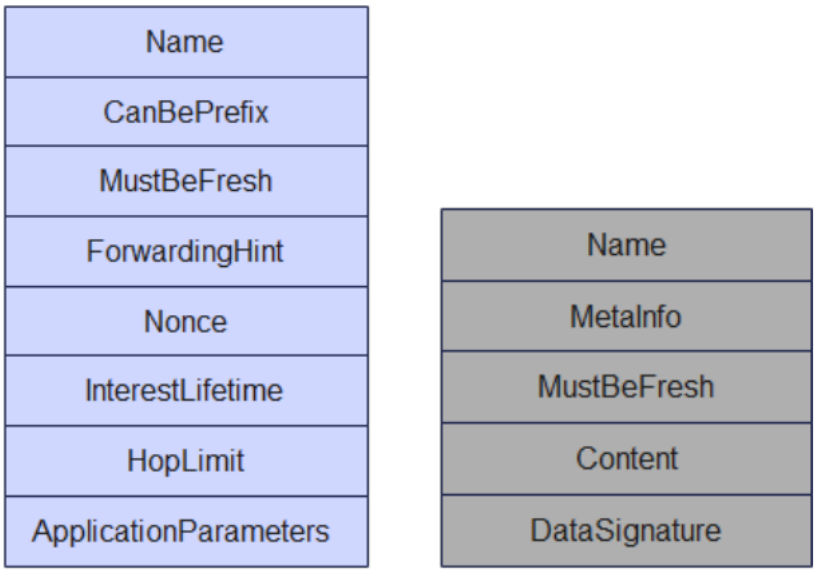

Figure 15. Interest and Data packets in Version 0.3 packet spec.

CSCA knows well its highest impacting attack in the NDN architecture because of what it deals with, for the exposure of the confidential and non-confidential contents relies on the CS cache. This attack is investigated by several researchers to make an efficient detection and mitigation mechanism.

Ertugrul Dogruluk et al. [38] introduced a mitigation technique, which basically stands by comparing, in each time step, the values obtained from the simulation by the predefined threshold for each of the metrics mentioned below:

- Current RTT.
- RTT threshold.
- Old RTT threshold.
- Cache hit ratio.
- Cache hit ratio threshold.
- Old cache hit threshold.

In case of the detection of an attack, the router applies CS random caching strategy, as explained in the flowchart of the figure 16.

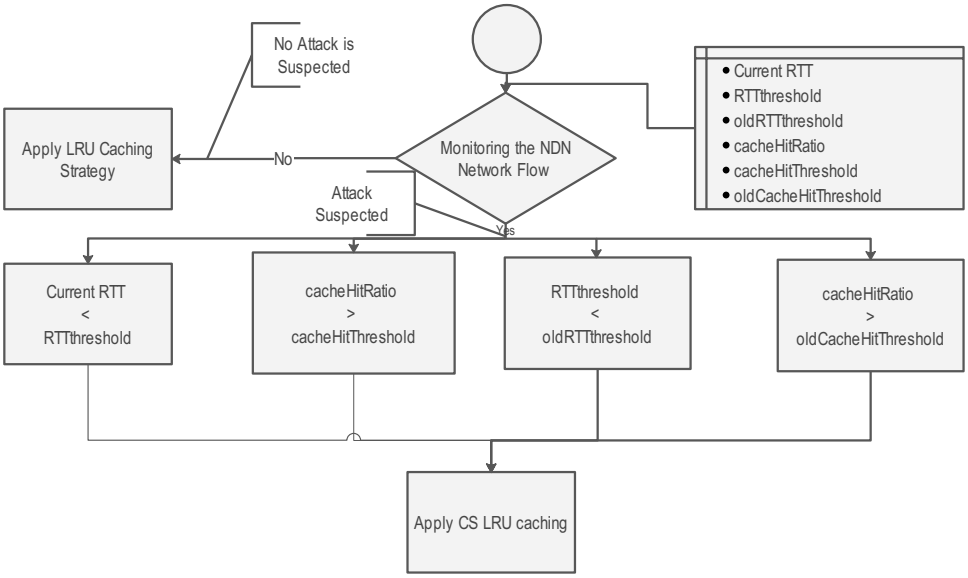

Figure 16. Statistical approach of detecting and mitigating CSCA.

Naveen Kumar et al. suggested in [81] a mechanism based on "Triggered Delay". This mechanism stands initially on the reception of an interest packet. The router examines the pattern name and compares it with the already stored pattern names. If the content is not found, then the PIT adds a new entry with two fields: "name" and "time". In the meantime, the field name is going to store the first component that received the interest packet from it, and the time field will contain the current time of receiving the certain interest.

The mechanism is based on adding a couple of fields in addition to it, which are "isAttack" and "CounterCount (CC)" fields. If the CC field is equal to a pre-defined threshold, then the field isAttack going to be set to True. The technique relies on adding the delay each time a prefix name is suspected, and the forwarding strategy remains the same as the normal NDN forwarding technique.

Naveen Kumar et al. [81] extended their work by proposing a new IDS for CSCA; they call it "Namespace-Based Privacy (NBP)". This approach comes to solve the problem of high space usage and high processing time and CPU usage that has been induced by the "Content-Based Privacy (CBP)". The authors added two main packet fields in each of the interest and data packets:

- *Private:* is a boolean value which takes it to be true as an entry if the data associated is private. Where each transfer is under the same name, the space is considered confidential.
- *NumPrivate:* is an integer value that defines the number of the private components that existed in that nameSpace.

Figure 17 illustrates an example of an interest packet with the nameSpace equaled to $/com/FireBase/ConfUserList/U700/0x17$.

In this mechanism, a delay is needed to define the presence of the interest. If this delay associated to every NameSpace Component is different from the Avg delay of the pre-recoded normal state delays for the same content, then the attack is suspected.

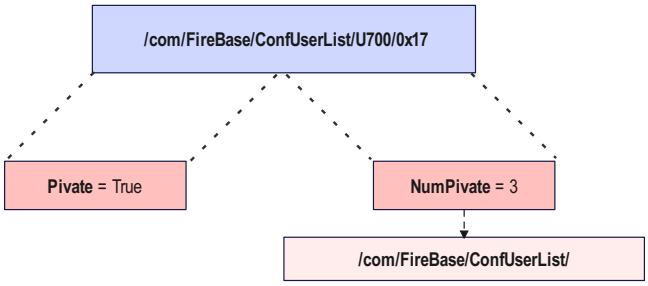

Figure 17. Namespace-based privacy (NBP) mechanism structure.

Vishwa Pratap Singh and R. L. Ujjwal [82], proposed a technique on how to assess the attack and detect the presence of the CSCA using the Attack Tree-Based technique. This basically uses subtrees from the network simulation that consisted of multiple nodes, such as:

- Data NameSpace.
- Interest NameSpace.
- Data packet.
- Interest packet.
- Content stored in the cache.
- The attack rate.

A Boolean algebra method has been proposed by the authors to identify the path where the attack comes from, the rate of the attack and the NameSpace associated to it.

Kyi Thar Ko et al. [83] suggest a method to protect the content, and relies on the cache using the Public Key Encryption with Keyword Search (PEKS)-Based NDN strategy. This mechanism stands on three main steps (Encrypt, Search and lookup, and Replay) with two main actors (the consumer and the CS cache).

This will force the attackers to request "null" values in their responses each time they request a content without the appropriate private key for that PrefixName.

Ertugrul Dogruluk et al. [84] proposed a new technique to protect VoNDN [85] against the CSCA by applying a model; they called it the DaD (Detection And Defence) Privacy model. The detection model is based on monitoring a couple of metrics, such as the Cache Hit Ratio (CHR), Cache Hit Ratio Threshold, and Average CHR.

This mitigation model is based on three main phases, such as:

- **Minor Phase:** Where a pre-defined threshold defined the Detection phase period window (TIME); if the attack detected superior to TIME, the face of where that interest is will be recorded.
- **Moderate Phase:** In this phase, the cache will change its caching strategy.
- **Severe Phase:** The top level alert, where the router will disable the caching in order to save the confidential content.

The authors in [39] suggested a solution against NDN Traffic Analysis, specifically the proactive attack. This relies on measuring the Hop-Count Delay $\Delta_{HopCount}$ and the NDN-RTT and compares it to a pre-defined threshold. Certain prefix pattern can be checked as a second solution suggested by the authors, and adding a traffic morphing [86] field in the NDN data packet.

Ertugrul Dogruluk et al. [87] extended their works into mitigating the CSCA attack, where they call it the Enhanced-DaD. The early solution (DaD) consisted of applying a statical condition in each phase to mitigate the attack. In order to improve the solution, the authors changed the conditions, where three conditions has been added to render the detection and mitigation dynamically effective:

- **If the value of** $CHR > 5\%$**:** Minor Phase is enabled for 3 s and the detection is kept in each 0.5 s.
- **If the attack state is still true:** Increment the Moderate phase for 3 s and keep the detection process for more than 0.5 s.
- **If the attack still on:** Severe phase is set as true and keeps discarding the caching process, where the NoCache strategy is set as the default policy of that NDN router.

Novel proposal has been made by Peng Kang et al. [88], where in order to mitigate the NDN network from the impact of CSCA, the Blockchain Content Processing has been made. A Smart Contract-Based, as the authors label it, poses a blockchain processing node.

Table 7 is summarizing most of the detection and mitigation mechanisms with the year of the release of each.

Table 7. Comparison of cache privacy attack detection and mitigation mechanisms.

Paper Reference	Year of Release	Type of Attack	Method
Lauinger et al.[77]	2012	TBA	Disabling Packet Fields-Based
Gao et al.[79]	2015	TBA	Pattern-Based
Ntuli et al.[78]	2018	TBA	Delay-Based
Naveen Kumar et al. [81]	2019	CSCA	NBP-Based
Ertugrul Dogruluk et al. [38]	2020	CSCA	Statical-Based
Naveen Kumar et al. [80]	2018	CSCA	Triggered Delay-Based
Vishwa Pratap Singh et al. [82]	2019	CSCA	Attack Tree-Based
Kyi Thar Ko et al. [83]	2020	CSCA	PEKS-Based
Ertugrul Dogruluk et al. [84]	2020	CSCA	DaD-Based
Alberto Compagno et al. [39]	2020	NTA	Delay, Pattern and Traffic Morphic-Based
Ertugrul Dogruluk et al. [87]	2022	CSCA	Enhanced-DaD-Based
Peng Kang et al. [88]	2022	CSCA	Blockchain-Based

5. Limits of Existing Attacks' Detection and Mitigation Techniques

5.1. Limits of Cache Pollution Attack Mitigation Mechanisms

As shown in Table 8, most of the detection mechanisms presented in Section 4 aim to provide a better strategy to prevent the effect of the attacks presented in Section 3 or to limit their impact. These mechanisms present a variety of vulnerabilities that can be exploited by the attackers to exhaust the router resources and expose the end-point node's identity, etc. An early solution, CacheShield [40] presented several limitations that affect the efficiency of the NDN routers, such as the complexity of the algorithm for detecting the malicious behavior of CPA. Another limitation is space exhaustion; since this mechanism does not consider the limited space that the cache of the CS has, the name placeholder can cause an issue at this point, in which this mechanism keeps storing them continuously. The prefix Hierarchy suggested by [42] solves the problem of exhausting the NDN router resources, but still depends on the topology itself. Moreover, this mechanism has issues when it comes to performing an attack consisting of caching the unpopular content of the popular prefix in order to confuse the mechanism. The mechanism, proposed by Guo et al. [41], has a medium level of memory exhaustion. In addition, this mechanism is not applicable on all types of topology, and keeps storing the information collected on each interest, which over-consumes the router space resources. The clustering method recommended by Yao et al. [44] has demonstrated good results in classifying different attack models such as LDA and FLA, but it still has a lack of certainty and makes too many false judgements, where in

this case the attacker keeps demanding the low popularity content to spoil the cache of the CS. ANFIS [47] has been a more reliable suggestion compared to previous mechanisms in terms of accuracy and efficiency. However, this mechanism may fall into two main states:

- The huge exhaustion of the resources of the router such as bandwidth, caching resources, etc., because many data packets need to be cached in order to decide the probability of the attack.
- The high rate of false positives leads the mechanism to allow content demanded by the malicious node to be cached, hindering the storage required by legitimate nodes.

Table 8. Cache pollution attack Detection mechanism limitations.

		Limitations								
	Ref	**Compromisable**	**Identity Leakage**	**Bandwidth Usage**	**Space Storage**	**CPU Overload**	**Accuracy**	**Topology**	**Complexity**	**False Positive**
CPA	[14]			✓	✓	✓			✓	
	[9]									✓
	[15]			✓	✓	✓	✓			
	[34]	✓								✓
	[55]	✓		✓	✓	✓				✓
	[50]					✓	✓	✓		
	[51]	✓			✓					✓
	[52]			✓		✓	✓	✓		✓
	[53]			✓	✓	✓		✓		

The mechanism proposed by Akanksha Gupa et al. [50] suffers from a couple of issues, such as the self-made simple topology, where the attack range can be different in wider topologies. In addition, the mechanism has less accuracy compared to the state-of-the-art mitigation mechanism. Cache nFace [51] falls into a couple of gaps, such as the exhaustion of the NDN router storage capacity. This mechanism can mitigate the attack on the NDN edge routers but is still unable to protect the intermediate routers; where the attackers can increase the hop count in order to avoid the detector router node, this will afford a low accuracy. Moreover, the mechanism did not take into consideration that low-popularity content can be requested by legitimate consumers. Ref. [52] has a low accuracy in detecting the attack in wider topology. In addition, this mechanism can reach the maximum CPU usage, where it keeps caching and removing from the cache the undesirable content.

AHISM-B [53] is optimal in terms of content delivery and content verification, where it prevents CPA attacks. However, this mechanism falls into the gap of extreme exhaustion of the NDN routers' resources. The mechanism can also be escaped by attacks such as the Block withholding attack methods, 51% attack, Pool Hopping attack, etc. [89].

5.2. Limits of Interest Flooding Attack Mitigation Mechanisms

As shown in Table 9, the IFA detection mechanisms falls into multiple drawbacks. For instance, the mechanism presented in [63]. This mechanism relies on the edge NDN routers, and the authors do not take into consideration that those routers may become affected by the malicious nodes. False positive decisions may appear as legitimate consumers requesting unavailable content or low-popularity content. The authors of [65] proposed a detection solution based on a neural network, which may cause huge space exhaustion and high CPU usage to the NDN router. The authors, in [34], rely on the stored records of the previous upcoming interests, where the attackers can avoid the mitigation mechanism by launching the attack in the first early stage of running the network. In [23], the mechanism

is greedy in terms of the NDN routers' resource consumption, such as the storing capacity and CPU usage base. Authors in [57,58] consider a simple topology and they limited their IDS to a small sample, where the attack can be triggered from multiple sources on the same time.

As introduced by the authors in [59], the mechanisms in [60–62] rely on the continuous measuring of the upcoming interests; these obtained values are stored in the NDN router, where the capacity of the previously mentioned component can be affected in terms of the storing capacity and CPU overload. In [63], the simulation is performed based on a normal simple tree topology where the attack can come from different sources. In addition, the sub-trees need to be stored in the NDN routers, which can cause an overload in the storage capacity; in addition to that, the value given as the predefined threshold can be critical, and these values can be designed to punish legitimate consumers' requests. In the same case of [64], it is where the proposed mechanism used a simple tree topology without considering the CIFA application.

Finally, the solution presented in [65] needs an initialization phase in order to compare the obtained values, where the attack in this case can target the NDN routers in the first early stage of the simulation, which leads to a false judgment.

Table 9. Interest flooding attack detection mechanism limitations.

		Limitations								
	Ref	**Compromisable**	**Identity Leakage**	**Bandwidth Usage**	**Space Storage**	**CPU Overload**	**Accuracy**	**Topology**	**Complexity**	**False Positive**
Interest Flooding Attack (IFA)	[63]	✓								✓
	[65]			✓	✓	✓	✓			✓
	[34]						✓			✓
	[23]			✓	✓	✓				
	[57]							✓		✓
	[58]							✓		✓
	[59]							✓	✓	✓
	[60]							✓	✓	✓
	[61]							✓	✓	✓
	[62]							✓	✓	
	[63]	✓					✓	✓	✓	✓
	[64]	✓								✓
	[65]	✓								✓

5.3. Limits of Cache Poisoning Attack Mitigation Mechanisms

Various mitigation mechanisms that are presented in the last section poses numerous limitations that are summarized in Table 10. To mitigate the Cache Poisoning Attacks accurately, [60,61] use a Self-certifying-based method. This method presents several drawbacks that may lead to false positive decisions. Based on this point, these methods are focusing more on the static content, which means that the dynamic content cannot be detected. The above-mentioned mechanisms suffer from the extreme exhaustion of the NDN router, which can damage the main components' functionalities.

The authors in [68] introduced a technique based on the trust method. The main flaw of this technique is that even the attacker can be misclassified as a trusted node, where the intermediate NDN routers can be compromised or pre-configured to be serving malicious content.

As for [69,70], they fall into the gap of the false positive.In these two methods, the mechanism relies on the feedback of each neighbor NDN router node. This gives the

attacker the opportunity to empower his attack using one of the NDN routers as the ones that serve malicious content.

In [71–73,75], the main limits of the proposed mechanisms are the high CPU usage and space storage in the NDN routers.

The mechanism in [74] suffers from a false judgment, where the malicious NDN router can represent itself as a legitimate content carrier, and it gives high popularity to the malicious content. Same for the mechanism in [76], where the malicious consumers can be presented as VIP-Consumers and keep demanding the malicious content urgently. In this case, a trust value for each consumer needs to be calculated and verified in each content exchange.

Table 10. Cache poisoning attack detection mechanism limitations.

		Limitations								
	Ref	**Compromisable**	**Identity Leakage**	**Bandwidth Usage**	**Space Storage**	**CPU Overload**	**Accuracy**	**Topology**	**Complexity**	**False Positive**
Cache Poisoning Attack	[60]	✓								✓
	[61]	✓				✓				✓
	[68]	✓								✓
	[70]									✓
	[69]									✓
	[71]			✓	✓	✓			✓	
	[72]			✓	✓	✓			✓	
	[73]			✓	✓	✓			✓	
	[75]			✓	✓	✓			✓	
	[74]									✓
	[76]	✓								✓

5.4. Limits of Cache Privacy Attack Mitigation Mechanisms

Cache privacy attacks present a great challenge for researchers to invent new mechanisms that can resist them. Unfortunately, those mechanisms still have many limitations that are summarized in Table 11. The authors in [34] suggest disabling two main fields in the NDN packet specification [91,92] as shown in Figure 15 in order to mitigate the privacy of the content presented in the cache, such as for *"Paid-Content"*. Thus, by eliminating the scope field from the new packet specification, the *"hop-Limit"* field can still be manipulated as a scope field. As for the mechanisms presented in [23,57,58], which propose applying an extra delay on the suspected interest with a specific prefix, this would delay the attack such as in the case of TBA, but it still cannot resist if the intruder obtains the desired content.

Although the NDN-RTT presented in [38] is a good way to measure the presence of the CSCA attack, this attack cannot be totally mitigated by applying a dynamic caching switching strategy.

As for [80,81], the authors added different fields to detect the source of the attack, but on the other hand, the attack can be more severe if those fields are used to identify the consumers who request certain contents.

In [82], the authors suggested a method on how to detect the path of the attack, but no mitigation mechanism is suggested. Where in [83], the authors rely on adding a new cryptography signature that can consume the CPU usage of the limited-resourced NDN router even further.

As already presented in the mitigation mechanism DaD [84] and Enhanced-DaD [87], this mitigation mechanism relies on applying different caching strategies in order to avoid

falling into the CSCA. Where in this case, as shown in previous mitigation mechanisms, this technique will delay the attack but not mitigate it.

As shown in [39], the attack identification will consume a lot of resources in the NDN routers, especially by adding the new field, Traffic Morphing.

In the same consequences, the proposal of [88] uses a lot of the CPU usage and storage capacity in each NDN Router, especially when constructing each BlockChain block and verifying it.

Table 11. Cache Privacy Attack Detection mechanism limitations.

	Ref	Limitations								
		Compromisable	**Identity Leakage**	**Bandwidth Usage**	**Space Storage**	**CPU Overload**	**Accuracy**	**Topology**	**Complexity**	**False Positive**
Cache Privacy Attack	[34]	✓	✓				✓			
	[23]		✓					✓		✓
	[57]	✓					✓			
	[58]	✓								
	[38]						✓			✓
	[80]	✓								
	[81]	✓								
	[82]						✓			✓
	[83]			✓	✓	✓				
	[84]						✓			✓
	[87]						✓			✓
	[39]			✓	✓	✓				
	[88]				✓	✓				

6. NDN Security Open Research Issues

Multiple mitigation mechanism has been suggested in order to cover the impact of NDN security attacks discussed in previous sections. Multiple limitations have been identified in each mitigation mechanism model. In this section, we discuss the open research issues for each attack in terms of: the metrics that need to be used, the topology, the NDN router resource consumption, the technology model used, etc.

6.1. Open Research Issues for Cache Pollution Attacks

1. More efficient and accurate metrics need to be chosen in order to avoid the high false positive alarms.
2. NDN router resources need to be taken into consideration, such as CPU usage and Content space usage in order to build appropriate detection and mitigation mechanisms.
3. The mechanisms that rely on datasets, supervised learning techniques and offline simulations need to consider a real wider topology in order to be accurate in defining the exact detection values and the qualified thresholds.
4. The mitigation mechanisms need to take the right action according to the performance of NDN network status.
5. The right feature selection needs to be used in the detection mechanisms that are based on the Neural Network Approaches.

6.2. Open Research Issues for Interest Flooding Attacks

1. Static data as well as the dynamic requested data need to be taken into consideration in order to design the appropriate mitigation mechanism.

2. The phase of initialization can be critical in the identification of the attack, where the attack can be launched from the beginning of the simulation.
3. The proposed mitigation technique need not inflict high damage to NDN routers in terms of resource consumption.
4. Monitoring different NDN routers along the data transmission is needed. In addition, the hop count field can help to detect which NDN router can be attacked. The hop count field needs to be controlled in the detection mechanism.

6.3. Open Research Issues for Cache Poisoning Attacks

1. The attack can be realized by single/multiple malicious consumers or a combination of malicious consumers with malicious NDN routers. The mitigation mechanism needs to take all the attacking components into consideration.
2. A distributed detection system needs to be implemented in a large number of nodes in order to obtain the appropriate decision.
3. The appropriate metrics related to the CS cache need to be taken into consideration.
4. The detection based on the prefix variation should be able to differentiate the architecture of various prefix patterns.

6.4. Open Research Issues for Cache Privacy Attacks

1. Adding different fields can be critical in designing the mitigation mechanism, where the identification of the consumers anywhere else can create a conflict with the NDN privacy conservation.
2. The detection mechanisms that are based on adding an extra delay on the attacking request need to take into consideration the dynamic delay of requesting malicious contents, which means the predefined threshold needs to be accurate.
3. The low NDN routers resources need to be managed and taken into consideration.
4. A communication mechanism between the different nodes needs to be designed in order to protect the neighbor NDN routers as soon as any of the routers detects the attack, such as with an announcement in every predefined time step.
5. Boosting techniques in detecting the attack based on Neural Networks need to be used, similar to the case of [90].

7. Conclusions

The success of NDN has always been based on its security and its high performance, ensuring it a promising future in the network revolution. However, since NDN is not vulnerable to a range of basic attacks that are effective on the TCP/IP model, several new attacks have appeared that target NDN's main components. In this survey, we illustrated the potential threats that could affect the NDN architecture and classified them into four main classes, namely Cache Pollution Attacks, Interest Flooding Attacks, Cache Poisoning Attacks and Cache Privacy Attacks. Most of these vulnerabilities result from the availability of the data in the cache of intermediate routers. Moreover, we presented and discussed a variety of attack detection and mitigation mechanisms and classified them into different categories. Finally, we presented and identified the limits of the existing mechanisms to highlight the challenges that may be considered by the research community when designing efficient security solutions in NDN.

Although this survey focuses on the highly impactful attacks that can target the NDN architecture, the state-of-the-art solutions and their limitations with open research issues on each, we aim to extend our work into a better solution that take into consideration the issues presented in the recent mitigation mechanisms to develop an intelligent mechanism that can insure the collaborative communication between the nodes and the resource management in each NDN component. Moreover, we aim to provide a secured data transfer between the NDN nodes by respecting the security semantic of the NDN architecture.

Author Contributions: Conceptualization, A.H., N.H., H.T., M.H. and P.M.; Resources, A.H., N.H., H.T., M.H. and P.M.; Writing–original draft, A.H.; Writing–review editing, A.H., N.H., H.T., M.H.

and P.M.; Supervision, N.H., H.T., M.H. and P.M.; Project management, A.H., N.H., H.T., M.H. and P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research work has been carried out jointly at the IRT SystemX, IReSCoMath Research Lab and INRIA of Paris in the scope of the project EXPLO, which has received funding from the IRT SystemX.

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

Abbreviations

The following abbreviations are used in this manuscript:

PIT	Pending Interest Table
CS	Content Store
FIB	Forwarding Information Base
CPA	Cache Pollution Attack
IFA	Interest Flooding Attack
CDN	Content Delivery Network
P2P	Peer-to-Peer
DDB	Distributed Database
LRU	Least Recently Used
LFU	Least Frequently Used
FLA	False Locality Attack
LDA	Locality disruption attack
CHR	Cache Hit Ratio
ARD	Average Retreival Delay
HDR	Hit Damage Ratio
BIFA	Basic Interest Flooding Attack
CIFA	Collusive Interest Flooding Attack
SCAN	Smart Collaborative Attack in NDN
TBA	Timing-Based Attack
ODA	Object Discovery Attack
DFCA	Data Flow Cloning Attack
CSCA	Cache Side Channel Attack
NDN-RTT	NDN Round Trip Time
NTA	NDN Traffic Analysis Attack
CPMH	Cache Protection Method based on prefix Hierarchy
RVP	Request rate Variation for each Prefix
CV	Coefficient of Variation
ICAN	Intrusion detection system for CPA attack in NDN
ANFIS	Adaptive Neuro-Fuzzy Inference System
EWMA	Exponentially Weighted Moving Average
CUSUM	Cumulative Sum
HNIMFA	Heterogeneous N-Intertwined Mean Field Approximation
SIS	Susceptible-Infected-Susceptible
SVM	Support Vector Machine
JS	Jensen-Shannon
SDN	Software Defined Networking

References

1. Bhowmik, S. *Cloud Computing*; Cambridge University Press: Cambridge, UK, 2017. https://doi.org/10.10.1017/9781316941386.
2. Barkai, D. Technologies for sharing and collaborating on the Net. In Proceedings of the First International Conference on Peer-to-Peer Computing, Linköping, Sweden, 27-29 August 2001. https://doi.org/10.10.1109/P2P.2001.990419.
3. Velmurugan, L.; Manoharan, S. Designing Factors of Distributed Database System: A Review. *Data Min. Knowl. Eng.* **2020**, *12*, 7–10.
4. Cheriton, D.; Gritter, M. TRIAD: A New Next-Generation Internet Architecture. Available online: http://www-dsg.stanford.edu/triad/ (accessed on 1 December 2022).
5. Koponen, T.; Chawla, M.; Chun, B.-G.; Ermolinskiy, A.; Kim, K.H.; Shenker, S.; Stoica, I. A data-oriented (and beyond) network architecture. *SIGCOMM Comput. Commun. Rev.* **2007**, *37*, 181–192. https://doi.org/10.1145/1282380.1282402.

6. Trossen, D. Pursuit Project. Available online: http://www.fp7-pursuit.eu/PursuitWeb/ (accessed on 1 December 2022).
7. Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.; Briggs, N.; Braynard, R. Networking named content. *Commun. ACM* **2012**, *55*, 117–124. https://doi.org/10.1145/2063176.2063204.
8. Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J. D., Smetters, D. K., Papadopoulos, C. (2010). Named data networking (ndn) project. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC, 157, 158.
9. Tourani, R.; Misra, S.; Mick, T.; Panwar, G. Security, Privacy, and Access Control in Information-Centric Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 566–600. https://doi.org/10.1109/comst.2017.2749508.
10. Kumar, N.; Singh, A.K.; Aleem, A.; Srivastava, S. Security Attacks in Named Data Networking: A Review and Research Directions. *J. Comput. Sci. Technol.* **2019**, *34*, 1319–1350. https://doi.org/10.1007/s11390-019-1978-9.
11. Im, H.; Kim, D. An Overview of Content Poisoning in NDN: Attacks, Countermeasures, and Direction. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 2904–2918. https://doi.org/10.3837/tiis.2020.07.010.
12. Arulkumaran, G.; Rajalakshmi, N.R. Named Data Networking (NDN), Internet Architecture Design and Security Attacks. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 1281–1284. https://doi.org/10.35940/ijitee.k1258.09811s19.
13. Benmoussa, A.; Kerrache, C.A.; Lagraa, N.; Mastorakis, S.; Lakas, A.; Tahari, A.E.K. Interest Flooding Attacks in Named Data Networking: Survey of Existing Solutions, Open Issues, Requirements and Future Directions. *ACM Comput. Surv.* **2022**. https://doi.org/10.1145/3539730.
14. Lee, R.-T.; Leau, Y.-B.; Park, Y.J.; Anbar, M. A Survey of Interest Flooding Attack in Named-Data Networking: Taxonomy, Performance and Future Research Challenges. *IETE Tech. Rev.* **2021**, 1–19. https://doi.org/10.1080/02564602.2021.1957029.
15. Shah, M.S.M.; Leau, Y.-B.; Yan, Z.; Anbar, M. Hierarchical Naming Scheme in Named Data Networking for Internet of Things: A Review and Future Security Challenges. *IEEE Access* **2022**, *10*, 19958–19970. https://doi.org/10.1109/access.2022.3151864.
16. Jeet, R.; Kumar, P.A.R. A survey on interest packet flooding attacks and its countermeasures in named data networking. *Int. J. Inf. Secur.* **2022**, *21*, 1163–1187. https://doi.org/10.1007/s10207-022-00591-w.
17. Ali, Z.; Shah, M.A.; Almogren, A.; Din, I.U.; Maple, C.; Khattak, H.A. Named Data Networking for Efficient IoT-based Disaster Management in a Smart Campus. *Sustainability* **2020**, *12*, 3088. https://doi.org/10.3390/su12083088.
18. Khan, O.A.; Shah, M.A.; Din, I.U.; Kim, B.-S.; Khattak, H.A.; Rodrigues, J.J.P.C.; Farman, H.; Jan, B. Leveraging Named Data Networking for Fragmented Networks in Smart Metropolitan Cities. *IEEE Access* **2018**, *6*, 75899–75911. https://doi.org/10.1109/access.2018.2882811.
19. Quevedo, J.; Corujo, D. Selective Content Retrieval in Information-Centric Networking. *Sensors* **2022**, *22*, 8742. https://doi.org/10.3390/s22228742.
20. Karim, F.A.; Aman, A.H.M.; Hassan, R.; Nisar, K.; Uddin, M. Named Data Networking: A Survey on Routing Strategies. *IEEE Access* **2022**, *10*, 90254–90270. https://doi.org/10.1109/access.2022.3201083.
21. Mejri, S.; Touati, H.; Malouch, N.; Kamoun, F. Hop-by-Hop Congestion Control for Named Data Networks. In Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October–3 November 2017; pp. 114–119, https://doi.org/10.10.1109/AICCSA.2017.36.
22. Mejri, S.; Touati, H.; Kamoun, F. Hop-by-hop interest rate notification and adjustment in named data networks. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018; pp. 1–6, https://doi.org/10.10.1109/WCNC.2018.8377374.
23. Wang, K.; Guo, D.; Quan, W. Analyzing NDN NACK on Interest Flooding Attack via SIS Epidemic Model. *IEEE Syst. J.* **2019**, *14*, 1862–1873. https://doi.org/10.1109/jsyst.2019.2923841.
24. Nguyen, T.; Mai, H.-L.; Cogranne, R.; Doyen, G.; Mallouli, W.; Nguyen, L.; El Aoun, M.; De Oca, E.M.; Festor, O. Reliable Detection of Interest Flooding Attack in Real Deployment of Named Data Networking. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2470–2485. https://doi.org/10.1109/tifs.2019.2899247.
25. da Silva, E.T.; de Macedo, J.M.H.; Costa, A.L.D. NDN Content Store and Caching Policies: Performance Evaluation. *Computers* **2022**, *11*, 37. https://doi.org/10.3390/computers11030037.
26. Chatterjee, T.; Ruj, S.; Das Bit, S. Security Issues in Named Data Networks. *Computer* **2018**, *51*, 66–75. https://doi.org/10.1109/mc.2018.1151010.
27. Zhang, Z.; Wong, S.Y.; Shi, J.; Pesavento, D.; Afanasyev, A.; Zhang, L.On Certificate Management in Named Data Networking. *arXiv* **2020**, arXiv:2009.09339.
28. Bouk, S.H.; Ahmed, S.H.; Hussain, R.; Eun, Y. Named Data Networking's Intrinsic Cyber-Resilience for Vehicular CPS. *IEEE Access* **2018**, *6*, 60570–60585. https://doi.org/10.10.1109/ACCESS.2018.2875890.
29. Gasti, P.; Tsudik, G.; Uzun, E.; Zhang, L. DoS and DDoS in Named Data Networking. In Proceedings of the 2013 22nd International Conference on Computer Communication and Networks (ICCCN), Nassau, Bahamas, 30 July–2 August 2013; pp. 1–7. https://doi.org/10.10.1109/ICCCN.2013.6614127.
30. Al-Musawi, B.; Branch, P.; Armitage, G. BGP Anomaly Detection Techniques: A Survey. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 377–396. https://doi.org/10.10.1109/COMST.2016.2622240.
31. Deng, L.; Gao, Y.; Chen, Y.; Kuzmanovic, A. Pollution attacks and defenses for Internet caching systems. *Comput. Netw.* **2008**, *52*, 935–956. https://doi.org/10.10.1016/j.comnet.2007.11.019.

32. Hidouri, A.; Hadded, M.; Hajlaoui, N.; Touati, H.; Muhlethaler, P. Cache pollution attacks in the NDN architecture: Impact and analysis. In Proceedings of the 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia, 23–25 September 2021. http://dx.doi.org/10.23919/softcom52868.2021.9559049.
33. Buragohain, M.; Nandi, S. Demystifying security on NDN: A survey of existing attacks and open research challenges . In *The "Essence" of Network Security: An End-to-End Panorama*; Springer: Singapore, 2020; pp. 241–261. http://dx.doi.org/10.1007/978-981-15-9317-8_10.
34. Al-Share, R.A.; Shatnawi, A.S.; Al-Duwairi, B. Detecting and Mitigating Collusive Interest Flooding Attacks in Named Data Networking. *IEEE Access* **2022**, *10*, 65996–66017. https://doi.org/10.10.1109/ACCESS.2022.3184304.
35. Wu, Z.; Feng, W.; Lei, J.; Yue, M. I-CIFA: An improved collusive interest flooding attack in named data networking. *J. Inf. Secur. Appl.* **2021**, *61*, 102912. https://doi.org/10.1016/j.jisa.2021.102912.
36. Buragohain, M.; Kathar, C.J.; Kachari, C.; Nandi, S.K.; Nandi, S. SCAN: Smart Collaborative Attack in Named Data Networking. In Proceedings of the 2020 IEEE 45th Conference on Local Computer Networks (LCN), Sydney, Australia, 16–19 November 2020; pp. 124-133. https://doi.org/10.10.1109/LCN48667.2020.9314807.
37. Lauinger, T.; Laoutaris, N.; Rodriguez, P.; Strufe, T.; Biersack, E.; Kirda, E. *Privacy Implications of Ubiquitous Caching in Named Data Networking Architectures*; Technical Report; Northeastern University: Boston, MA, USA, 2012. https://tobias.lauinger.name/papers/ccn-cache-attackstr-iseclab-0812-001.pdf.
38. 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: Berlin/Heidelberg, Germany, 2018; pp. 33–46. http://dx.doi.org/10.1007/978-3-319-94421-0_3.
39. 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. http://dx.doi.org/10.1109/noms47738.2020.9110318.
40. Xie, M.; Widjaja, I.; Wang, H. Enhancing cache robustness for content-centric networking. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012. http://dx.doi.org/10.1109/infcom.2012.6195632.
41. Guo, H.; Wang, X.; Chang, K.; Tian, Y. Exploiting path diversity for thwarting pollution attacks in named data networking. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2077–2090. https://doi.org/10.1109/tifs.2016.2574307.
42. Kamimoto, T.; Mori, K.; Umeda, S.; Ohata, Y.; Shigeno, H. Cache protection method based on prefix hierarchy for content-oriented network. In Proceedings of the 2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2016. http://dx.doi.org/10.1109/ccnc.2016.7444816.
43. Zhang, G.; Liu, J.; Chnag, X.; Chen, Z. Combining Popularity and Locality to Enhance In-Network Caching Performance and Mitigate Pollution Attacks in Content-Centric Networking. *IEEE Access* **2017**, *5*, 19012–19022. https://doi.org/10.10.1109/ACCESS.2017.2754058.
44. Yao, L.; Fan, Z.; Deng, J.; Fan, X.; Wu, G. Detection and defense of cache pollution attacks using clustering in named data networks. *IEEE Trans. Dependable Secur. Comput.* **2020**, *17*, 1310–1321. https://doi.org/10.1109/tdsc.2018.2876257.
45. Hidouri, A.; Touati, H.; Hadded, M.; Hajlaoui, N.; Muhlethaler, P. A detection mechanism for cache pollution attack in named data network architecture. In *Advanced Information Networking and Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 435–446. http://dx.doi.org/10.1007/978-3-030-99584-3_38.
46. Park, H.; Widjaja, I.; Lee, H. Detection of cache pollution attacks using randomness checks. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012. http://dx.doi.org/10.1109/icc.2012.6363885.
47. Karami, A.; Guerrero-Zapata, M. An ANFIS-based cache replacement method for mitigating cache pollution attacks in Named Data Networking. *Comput. Netw.* **2015**, *80*, 51–65. https://doi.org/10.1016/j.comnet.2015.01.020.
48. Kumar, N.; Srivast, S. *IBPC: An Approach for Mitigation of Cache Pollution Attack in NDN using Interface-Based Popularity*; Research Square Platform LLC, Durham, North Carolina 2021. http://dx.doi.org/10.21203/rs.3.rs-682924/v1.
49. Lei, K.; Fang, J.; Zhang, Q.; Lou, J.; Du, M.; Huang, J.; Wang, J.; Xu, K. Blockchain-Based cache poisoning security protection and privacy-aware access control in NDN vehicular edge computing networks. *J. Grid Comput.* **2020**, *18*, 593–613. https://doi.org/10.1007/s10723-020-09531-1.
50. Gupta, A.; Nahar, P. Detection of Cache Pollution Attacks in a Secure Information-Centric Network. In *Data Analytics and Management*, Khanna, A., Gupta, D., Pólkowski, Z., Bhattacharyya, S., Castillo, O., Eds.; Lecture Notes on Data Engineering and Communications Technologies; Springer, Singapore, 2021, Volume 54. https://doi.org/10.1007/978-981-15-8335-3_30.
51. Nasserala, A.; Bastos, I.V.; Monteiro Moraes, I. Cache nFace: A simple countermeasure for the producer-consumer collusion attack in Named Data Networking. *Ann. Telecommun.* **2019**, *74*, 125–137. https://doi.org/10.1007/s12243-018-0669-9.
52. Zhou, J.; Luo, J.; Deng, L.; Wang, J. Cache Pollution Prevention Mechanism Based on Cache Partition in V-NDN. In Proceedings of the 2020 IEEE/CIC International Conference on Communications in China (ICCC), Chongqing, China, 9–11 August 2020; pp. 330–335. https://doi.org/10.10.1109/ICCC49849.2020.9238838.
53. Li, B.; Ma, M. An Advanced Hierarchical Identity-Based Security Mechanism by Blockchain in Named Data Networking. *J. Netw. Syst. Manag.* **2023**, *31*. https://doi.org/10.1007/s10922-022-09689-x.

54. Dai, H.; Wang, Y.; Fan, J.; Liu, B. Mitigate DDoS attacks in NDN by interest traceback. In Proccedings of the 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Turin, Italy, 14–19 April 2013. http://dx.doi.org/10.1109/infcomw.2013.6970722.
55. Compagno, A.; Conti, M.; Gasti, P.; Tsudik, G. Poseidon: Mitigating interest flooding DDoS attacks in Named Data Networking. In Proccedings of the 38th Annual IEEE Conference on Local Computer Networks, Sydney, Australia, 21–24 October 2013. http://dx.doi.org/10.1109/lcn.2013.6761300.
56. Karami, A.; Guerrero-Zapata, M. A hybrid multiobjective RBF-PSO method for mitigating DoS attacks in Named Data Networking. *Neurocomputing* **2015**, *151*, 1262–1282. https://doi.org/10.1016/j.neucom.2014.11.003.
57. Zhi, T.; Liu, Y.; Wang, J.; Zhang, H. Resist Interest Flooding Attacks via Entropy–SVM and Jensen–Shannon Divergence in Information-Centric Networking. *IEEE Syst. J.* **2020**, *14*, 1776–1787. https://doi.org/10.10.1109/JSYST.2019.2939371.
58. Benmoussa, A.; Tahari A el, K.; Kerrache, C.A.; Lagraa, N.; Lakas, A.; Hussain, R.; Ahmad, F. MSIDN: Mitigation of Sophisticated Interest flooding-based DDoS attacks in Named Data Networking. *Future Gener. Comput. Syst.* **2020**, *107*, 293–306. https://doi.org/10.1016/j.future.2020.01.043.
59. Benarfa, A.; Hassan, M.; Losiouk, E.; Compagno, A.; Yagoubi, M.B.; Conti, M. ChoKIFA+: An early detection and mitigation approach against interest flooding attacks in NDN. *Int. J. Inf. Secur.* **2021**, *20*, 269–285. https://doi.org/10.1007/s10207-020-00500-z.
60. Alhisnawi, M.; Ahmadi, M. Detecting and Mitigating DDoS Attack in Named Data Networking. *J. Netw. Syst. Manag.* **2020**, *28*, 1343–1365. https://doi.org/10.1007/s10922-020-09539-8.
61. Wu, Z.; Feng, W.; Yue, M.; Xu, X.; Liu, L. Mitigation measures of collusive interest flooding attacks in named data networking. *Comput. Secur.* **2020**, *97*, 101971. https://doi.org/10.1016/j.cose.2020.101971.
62. Liu, L.; Feng, W.; Wu, Z.; Yue, M.; Zhang, R. The Detection Method of Collusive Interest Flooding Attacks Based on Prediction Error in NDN. *IEEE Access* **2020**, *8*, 128005–128017. https://doi.org/10.10.1109/ACCESS.2020.3008723.
63. Xing, G.; Chen, J.; Hou, R.; Zhou, L.; Dong, M.; Zeng, D.; Luo, J.; Ma, M. Isolation Forest-Based Mechanism to Defend against Interest Flooding Attacks in Named Data Networking. *IEEE Commun. Mag.* **2021**, *59*, 98–103. https://doi.org/10.10.1109/MCOM.001.2000368.
64. Zhang, X.; Li, R.; Hou, W. Attention-Based LSTM model for IFA detection in named data networking. *Secur. Commun. Netw.* **2022**, *2022*, 1–14. https://doi.org/10.1155/2022/1812273.
65. Wu, Z.; Peng, S.; Liu, L.; Yue, M. Detection of Improved Collusive Interest Flooding Attacks Using BO-GBM Fusion Algorithm in NDN. *IEEE Trans. Netw. Sci. Eng.* 2022, https://doi.org/10.10.1109/TNSE.2022.3206581.
66. Kim, D.; Nam, S.; Bi, J.; Yeom, I. Efficient content verification in named data networking. In Proceedings of the 2nd ACM Conference on Information-Centric Networking, San Francisco, CA, USA, 30 September–2 October 2015. http://dx.doi.org/10.1145/2810156.2810165.
67. Ghali, C.; Tsudik, G.; Uzun, E. Needle in a Haystack: Mitigating content poisoning in named-data networking. In Proceedings of the 2014 Workshop on Security of Emerging Networking Technologies. http://dx.doi.org/10.14722/sent.2014.23014.
68. Rezaeifar, Z.; Wang, J.; Oh, H. A trust-based method for mitigating cache poisoning in Name Data Networking. *J. Netw. Comput. Appl.* **2018**, *104*, 117–132. https://doi.org/10.1016/j.jnca.2017.12.013.
69. Konorski, J. Mitigating Time-Constrained Stolen-Credentials Content Poisoning in an NDN Setting. In Proceedings of the 2019 29th International Telecommunication Networks and Applications Conference (ITNAC), Auckland, New Zealand, 27–29 November 2019; pp. 1–7. https://doi.org/10.10.1109/ITNAC46935.2019.9077973.
70. Saha, B.K.; Misra, S. Mitigating NDN-Based Fake Content Dissemination in Opportunistic Mobile Networks. *IEEE Trans. Mob. Comput.* **2020**, *19*, 1375–1386. https://doi.org/10.10.1109/TMC.2019.2908161.
71. Baranski, S.; Konorski, J. Mitigation of Fake Data Content Poisoning Attacks in NDN via Blockchain. In Proceedings of the 2020 30th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, Australia, 25–27 November 2020; pp. 1–6. https://doi.org/10.10.1109/ITNAC50341.2020.9315048.
72. Singh, V.P.; Ujjwal, R.L. NDN Content Poisoning Attack Mitigation Using Fuzzy-Reputation Based Trust. In *Innovations in Cyber Physical Systems*, Singh, J., Kumar, S., Choudhury, U., Eds.; Lecture Notes in Electrical Engineering; Springer: Singapore, 2021, Volume 788. https://doi.org/10.1007/978-981-16-4149-7_29.
73. Hussain, S.; Ullah, S.S.; Gumaei, A.; Al-Rakhami, M.; Ahmad, I.; Arif, S.M. A Novel Efficient Certificateless Signature Scheme for the Prevention of Content Poisoning Attack in Named Data Networking-Based Internet of Things. *IEEE Access* **2021**, *9*, 40198–40215. https://doi.org/10.10.1109/ACCESS.2021.3063490.
74. Feng, M.; Li, R.; Hu, Y.; Yu, M. A Caching Strategy Based on Content Popularity Level for NDN. In *Advances in Artificial Intelligence and Security. ICAIS 2021*, Sun, X., Zhang, X., Xia, Z., Bertino, E., Eds.; Communications in Computer and Information Science; Springer: Cham, Germany, 2021, Volume 1424. https://doi.org/10.1007/978-3-030-78621-2_61.
75. Qureshi, A.M.; Anjum, N.; Rais RN, B.; Ur-Rehman, M.; Qayyum, A. Detection of malicious consumer interest packet with dynamic threshold values. *PeerJ Comput. Sci.* **2021**, *7*, e435. https://doi.org/10.7717/peerj-cs.435.
76. Hou, S.; Hu, Y.; Tian, L. Named data network dynamic cache placement strategy based on programmable data plane. *IET Networks*. **2022**. https://doi.org/10.1049/ntw2.12067.
77. 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.

78. Ntuli, N.; Han, S. Detecting router cache snooping in Named Data Networking. In Proceedings of the 2012 International Conference on ICT Convergence (ICTC), Jeju, Korea, 15–17 October 2012. http://dx.doi.org/10.1109/ictc.2012.6387155.
79. Gao, M.; Zhu, X.; Su, Y. Protecting router cache privacy in named data networking. In Proceedings of the 2015 IEEE/CIC International Conference on Communications in China (ICCC), Shenzhen, China, 2–4 November 2015. http://dx.doi.org/10.1109/iccchina.2015.7448754.
80. Kumar, N.; Singh, A.K.; Srivastava, S. A triggered delay-based approach against cache privacy attack in NDN. *Int. J. Netw. Distrib. Comput.* **2018**, *6*, 174. https://doi.org/10.2991/ijndc.2018.6.3.5.
81. Kumar, N.; Aleem, A.; Singh, A.K.; Srivastava, S. NBP: Namespace-based privacy to counter timing-based attack in named data networking. *J. Netw. Comput. Appl.* **2019**, *144*, 155–170. https://doi.org/10.1016/j.jnca.2019.07.004.
82. Singh, V.P.; Ujjwal, R.L. Privacy attack modeling and risk assessment method for name data networking. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2019; pp. 109–119. http://dx.doi.org/10.1007/978-981-13-6861-5_10.
83. Ko, K.T.; Hlaing, H.H.; Mambo, M. A peks-based NDN strategy for name privacy. *Future Internet* **2020**, *12*, 130. https://doi.org/10.3390/fi12080130.
84. 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. https://doi.org/10.1109/access.2020.3014898.
85. Ghasemi, C.; Yousefi, H.; Zhang, B. (2021). Internet-Scale video streaming over NDN. *IEEE Netw.* **2021**, *35*, 174–180. https://doi.org/10.1109/mnet.121.1900574.
86. Xu, Z.; Khan, H.; Muresan, R. TMorph: A Traffic Morphing Framework to Test Network Defenses Against Adversarial Attacks. In Proceedings of the 2022 International Conference on Information Networking (ICOIN), Jeju-si, Republic of Korea, 12–15 January 2022. https://doi.org/10.10.1109/ICOIN53446.2022.9687165.
87. 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.
88. Kang, P.; Wenzhong, Y.; Ding, T. Blockchain document forwarding and proof method based on NDN network. *IEEE Access* **2022**, *10*, 75312–75322. https://doi.org/10.1109/access.2022.3178992.
89. Chen, Y.; Chen, H.; Zhang, Y.; Han, M.; Siddula, M.; Cai, Z. A survey on blockchain systems: Attacks, defenses, and privacy preservation. *High-Confid. Comput.* **2022**, *2*, 100048. https://doi.org/10.10.1016/j.hcc.2021.100048.
90. Cao, B.; Li, C.; Song, Y.; Fan, X. Network Intrusion Detection Technology Based on Convolutional Neural Network and BiGRU. *Comput. Intell. Neurosci.* **2022**, *2022*, 1942847. https://doi.org/10.1155/2022/1942847.eCollection2022.
91. NDN Packet Format Specification—NDN Packet Format Specification 0.1 documentation. (n.d.). Available online: https://named-data.net/doc/NDN-packet-spec/0.1/ (accessed on 1 December 2022).
92. NDN Packet Format Specification 0.3. Named-Data.Net. 2022. Available online: https://named-data.net/doc/NDN-packet-spec/current/ (accessed on 1 December 2022).