Kumar N, Singh AK, Aleem A *et al.* Security attacks in named data networking: A review and research directions. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(6): 1319–1350 Nov. 2019. DOI 10.1007/s11390-019-1978-9

Security Attacks in Named Data Networking: A Review and Research Directions

Naveen Kumar, Ashutosh Kumar Singh, *Member*, *ACM*, *IEEE*, Abdul Aleem and Shashank Srivastava, *Member*, *ACM*, *IEEE*

Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology Allahabad Prayagraj 211004, India

E-mail: {nk10121989, ashuit89, er.aleem}@gmail.com; shashank12@mnnit.ac.in

Received September 4, 2018; revised September 8, 2019.

Abstract Contents such as audios, videos, and images, contribute most of the Internet traffic in the current paradigm. Secure content sharing is a tedious issue. The existing security solutions do not secure data but secure the communicating endpoints. Named data networking (NDN) secures the data by enforcing the data publisher to sign the data. Any user can verify the data by using the public key of the publisher. NDN is resilient to most of the probable security attacks in the TCP/IP model due to its new architecture. However, new types of attacks are possible in NDN. This article surveys the most significant security attacks in NDN such as interest flooding attacks, cache privacy attacks, cache pollution attacks, and content poisoning attacks. Each attack is classified according to their behavior and discussed for their detection techniques, countermeasures, and the affected parameters. The article is an attempt to help new researchers in this area to gather the domain knowledge of NDN. The article also provides open research issues that could be addressed by researchers.

Keywords named data networking (NDN), interest flooding attack, cache privacy attack, cache pollution attack, content poisoning attack

1 Introduction

Transmission Control Protocol/Internet Protocol (TCP/IP) model was developed as a solution to resource sharing and conversation among hosts. In the current paradigm, TCP/IP has to bear a significant amount of IP traffic load. As per the report of Cisco Visual Networking Index 2016, the global IP traffic per month is expected to reach 196 exabytes, and the IP video traffic is expected to be 82% of all traffic by 2020①. For dealing with the vast amount of IP traffic, many types of solutions (such as Content Delivery Network (CDN)[1], Peer to Peer (P2P)[2], and Distributed Database (DDB)[3]) are tried as an overlay. However, these solutions delay the transmission of packets due to the underlying network. This shortcoming demands a new network architecture that can be used for communication via the Internet. Information-Centric Networking (ICN)[4] overcomes this problem of the TCP/IP model. The most significant ICNs are DONA[5], COMET[6], CCN[7], and Named Data Networking (NDN)[8,9].

NDN is one of the most suitable candidates for future Internet architecture. In NDN, each data item has its unique human-readable hierarchical name like "*/bbc/news/morning/vidio.mp4*". This name is used for forwarding, routing, and fetching content. The in-network caching and the named-prefix based routing features of NDN expedite the delivery of packets. The in-network caching of NDN minimizes the delay by caching content within routers. The named-prefix based routing features of NDN reduce the DNS overhead involved in a TCP/IP communication.

Currently, NDN is running as an overlay over TCP/IP on a testbed② with NLSR[10] as the routing protocol. There are many NDN applications such

Survey

①https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html, Sept. 2019.

②https://named-data.net/ndn-testbed/, Sept. 2019.

as Chronoshare[11], Chronos[12], NDNFit[13], and Ndnrtc[14] which have been developed over Network Forwarding Daemon (NFD)[15].

NDN uses two types of packets, namely interest packet and data packet. The interest packet is used for resting the data packet. The data packet has the actual content which is sent as a response to the interest packet. NDN uses three data structures for forwarding packets — content store (CS), pending interest table (PIT), and forwarding information base (FIB). CS stores the data packets received by the router. PIT contains the interest packet entries and the corresponding list of incoming interfaces. FIB contains the list of named prefix and the ID of the interface through which the interest packet should be forwarded.

When the router receives an interest packet, the router performs a lookup in CS. If the data packet is found in CS, then it is replied through the interface which received the interest packet. Otherwise, the router looks into the PIT. If an entry is found in the PIT, then the ID of the interface which received the interest packet is concatenated with the interface list and stored as an update in the PIT. The entry in the PIT signifies that the packet has been forwarded earlier. In case the entry is not found in the PIT, a lookup is then performed in FIB for the outgoing interface, and an entry for the interest packet is created in the PIT. The interest packet is forwarded through the searched outgoing interface. If the entry is not found in FIB, then the interest packet is dropped, or a negative acknowledgment is sent depending on the router's policy.

When the router receives a data packet, the router first looks into PIT; if a matching entry is found then the data packet is forwarded to each interface, and the data packet is cached in CS. Otherwise, it is dropped.

Security solutions like Secure Sockets Layer (SSL) try to secure communicating endpoints (source or destination), but the data itself is not secure. In NDN, each data packet is signed by a publisher, and the consumer checks it using the publisher's public key. There is no identity of consumers through which an attacker can target a particular consumer. Therefore, NDN is resilient to most of the attacks which are possible in the TCP/IP model. However, new attacks are possible in NDN. The major attacks are the interest flooding attack, cache privacy attack, cache pollution attack, and content poisoning attack. Besides, there are also some security issues, such as name privacy, signature privacy, security in routing & forwarding and application level security, which are related to the trust between the consumer and the publisher applications. Zhang *et al.*[16] elaborated trust in NDN using an NDN application NDNFit[13]. In this trust model, each networked system (e.g., an organization, a smart home, or a cloud service provider) authority has its trust anchor(s) which can be discovered by underlying entities using their local system setting.

In interest flooding attack[17,18], the attacker sends a large number of non-existing interest packets to the network. Each router between the publisher and the attacker creates an entry for these packets in its PIT. As no data packet exists corresponding to the interest packets, each router is flooded with PIT entries. As a consequence, no space is left to handle interest packets sent by a legitimate consumer. In cache privacy attack[19,20], the attacker tries to find the recently cached data packets in the CS of the gateway router. Thus, the attacker gets the user's information about access patterns, interests, etc. that can be further used to perform other attacks. In cache pollution attack[21,22], the attacker sends interest packets to routers for the unpopular or rare data. This makes unpopular data cached which reduces the hit ratio of CS and thus affects the NDN performance. In content poisoning attack[23,24], the attacker tries to insert fake or corrupted data packets into CS. When normal users request data packets, then these poisoned data packets get spread in the CS of other routers.

Saxena *et al.*[25] discussed the different aspects of NDN such as routing, forwarding, security, and mobility. They classified various security attacks according to NDN layers. But, the security goals (confidentiality, integrity, and availability) affected by these attacks have been ignored. Chen and Mizero[26] focused on privacy aspects in NDN and put some open questions such as namespace management, interest flooding attack, and cache pollution attack. However, they did not discuss detection and mitigation approaches for security attacks.

This review article discusses security attacks in NDN which do not have better solutions yet. The security issues, such as name and signature privacy, the privacy of application, are based on the trust model. These issues have not been included in this article as they have been exhaustively discussed in [16]. Interested readers can refer to [16] for further details. This article mainly focuses on major security attacks, such as interest flooding attack, cache privacy attack, cache pollution attack, and content poisoning attack, which can happen in the presence of a trust model also.

The main contributions of this article are as follows:

• a critical analysis of four major security attacks in NDN vis-á-vis attackers, victims, compromised security goals and affected data structures;

• the demonstration of NDN's resilience to most of the attacks that are possible in the TCP/IP model;

• the analytical discussion for the countermeasures of the discussed security attacks in NDN along with their limitations;

• an assistance for new researchers in NDN by providing open research issues for all four attacks.

The rest of the article is organized as follows. Section 2 presents various security aspects of NDN and major attacks that are possible in NDN. Section 3 describes various characteristics of interest flooding attack such as its type, parameters of detection, the granularity of detection, and applied countermeasures along with the explanation of the limitations related to the countermeasures. Section 4 illustrates cache privacy attack, including its types, proposed countermeasures, and limitations. Section 5 covers various aspects of cache pollution attack such as its type, detection approaches, caching approaches, and distribution used for generating request packets. Section 6 describes content poisoning attack, including its types, detection approaches, applied countermeasures, and the entity (consumer or router) that detects the attack. Section 7 presents open research challenges in the NDN security. Finally, Section 8 concludes this article.

2 Security in Named Data Networking

Data structures used by the NDN router and forwarding pipeline have been discussed in Section 1. The important fields of interest packets and data packets and the security aspects of NDN are discussed in this section.

The packet formats for interest packets and data packets according to current NDN Packet Format Specification 0.3[③] are shown in Fig.1. The question mark (?) at the end of a field indicates that the field is not a compulsory field in interest or data packets. The description of fields in the interest packet is as follows.

2.1 Important Fields in Interest Packet

1) *Name*. This field specifies the name of the requested content in which a consumer is interested.

2) *CanBePrefix*. If the consumer sets this field, then the name of the received data packet can be either the prefix of the name of the interest packet or the same as the name of the interest packet. If this field is not set, then the name of the data packet is exactly the same as that of the interest packet.

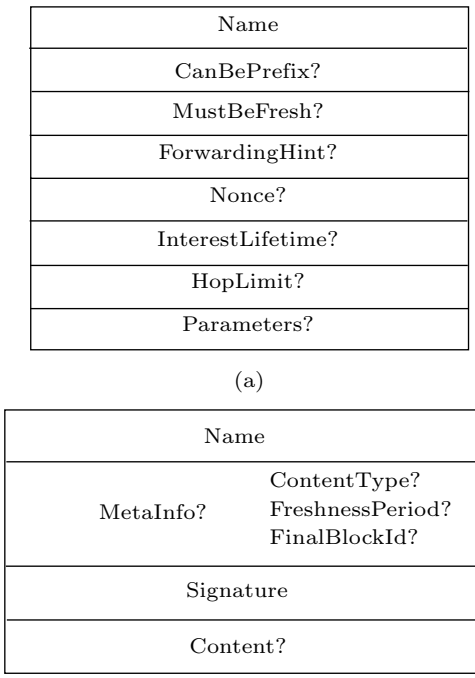

Fig.1. Packets in NDN[③]. (a) Interest packet. (b) Data packet.

3) *MustBeFresh*. The consumer sets this field to specify that the received data packet must not be stale.

4) *ForwardingHint*. The consumer sets this field to specify the forwarding paths of the interest packet.

5) *Nonce*. An interest packet is uniquely identified using the Name and Nonce combined, which in turns helps to detect looping interests.

6) *InterestLifetime*. This field indicates the approximate remaining lifetime of an interest packet.

7) *HopLimit*. This field indicates the number of hops that an interest packet is allowed to be forwarded.

8) *Parameters*. This field is used to parameterize the request for the data packet, and it contains any arbitrary data.

2.2 Important Fields in Data Packet

1) *Name*. This field specifies the name of the data packet, which is the same as the name of interest packet with a difference that it can sometimes have extra components as a suffix.

[③]https://named-data.net/doc/NDN-packet-spec/current/, Sept. 2019.

2) *MetaInfo*. This field contains the information about the data packet. It has three subfields as follows.

a) *ContentType*. This subfield specifies the type of the content.

b) *FreshnessPeriod*. This subfield specifies the time after which the content will be marked as "non-fresh" after being received by a router.

c) *FinalBlockId*. This subfield specifies the final block in a sequence of fragments.

3) *Content*. This field comprises of arbitrary bytes, which holds the actual content.

4) *Signature*. This field stores the signature, which is generated by hashing the entire data packet excluding the signature field and then encrypting the generated hash using the private key of the publisher.

2.3 Security Issues in Named Data Networking

The fundamental purpose of any network is to share resources such as pictures, texts, videos, and web content. Flawless communication is achieved through the implementation of security goals such as confidentiality, integrity, and availability. Confidentiality ensures that the data must be accessed only by the authorized person. Integrity implies the data received by the receiver must be the same as sent by the sender. Availability ensures that services provided by a network must be available for an authorized user.

Some popular attacks in the current TCP/IP model are snooping, traffic analysis attack, modification attack, masquerading, replay attack, repudiation, Denial of Service (DoS) attack, and Distributed Denial of Service (DDoS) attack. Table 1 shows the security goals breached by these attacks and their possibility in NDN. For example, from the first row of Table 1, it can be inferred that Snooping affects confidentiality only and is not possible in NDN.

In Snooping, an attacker tries to access confidential data of a user(s) and use it for his/her benefit. Snooping includes a wide range of attacks like observing e-mail of others or watching what someone else is typing. It is generally done through specialized software tools such as a keylogger, snoop server. Snooping in NDN is not possible due to the absence of the identity of the host. In a traffic analysis attack, an attacker monitors the user to predict patterns of usage of the Internet. The attacker can use the information gained from this attack for performing other attacks. Again, the absence of a host's identity like IP address makes it tough to perform this attack in NDN. However, the recently cached content can be predicted in NDN by observing the time period of reply.

In a modification attack, the attacker not only accesses the data but also tries to modify it. This attack is not possible in NDN as each piece of data is signed by the publisher, which can be verified by the consumer. However, if the router itself is malicious and modifies the data packet, then the consumer may receive a corrupted data packet. The consumer can only verify this corrupted data packet after receiving a valid publisher's public key.

In a masquerading attack, the attacker pretends to be someone else so that he/she can gain some useful information about the user. This attack is not possible in NDN as the publisher signs each piece of data by using its private key.

In a replay attack, the attacker tries to obtain a copy of the message from the sender and sends this message to the receiver. The receiver assumes that the intended sender has sent the received message. This attack is also not possible in NDN since the name and the nonce identify each interest packet. If the NDN router receives the same interest packet (having the same name and nonce), then the router will assume that the interest packet is replayed; therefore it should be dropped. Thus, NDN protects from replay attack from the network layer itself.

Table 1. Effect of Security Attacks on Security Goals

Attack	Security Goals Violated			Possible in NDN
	Confidentiality	Integrity	Availability	
Snooping	Yes	No	No	No
Traffic analysis attack	Yes	No	No	No
Modification attack	Yes	Yes	No	No
Masquerading	Yes	Yes	No	No
Replay attack	No	Yes	No	No
Repudiation	No	Yes	No	No
DoS attack	No	No	Yes	Yes
DDoS attack	No	No	Yes	Yes

In repudiation, an attacker may be the sender or receiver of the message. The sender may deny that he/she has sent the message and the receiver may deny that he/she has received the message. This attack is also not possible in NDN since there is a trust model between consumers and publishers.

In a DoS attack, the attacker tries to capture the bandwidth of the network by sending a large number of messages to a network or a host. This attack makes network resource such as bandwidth unavailable for legitimate users. It is hard to perform this attack on NDN as DoS requires the IP address of a host, but there is no IP address or any other identity of a host in NDN. However, the attacker can send a large number of non-existing interest packets to the network. These interest packets will create entries in the PIT of the intermediate routers, thus making PIT unavailable for the legitimate users. The DDoS attack is similar to DoS except that there are multiple entities involved in this attack. DDoS in TCP/IP is done through remotely controlled devices called bots. These bots use spoofed IP addresses to mask their true identities. Similar to DOS, DDoS can also be performed in NDN by multiple attackers.

From the discussion, it can be inferred that NDN is resilient to most of the attacks that are possible in TCP/IP. However, NDN is vulnerable to new types of attacks, which will be discussed in Subsection 2.4.

2.4 Security Attacks in NDN

There are four major security attacks in NDN as shown in Fig.2. Apart from the major attacks, there are some other security issues, such as name privacy and signature privacy, which could be attacked. These attacks do not come under the scope of this article and have been assumed to be included in the miscellaneous category. Each of the four major attacks will be briefed next and discussed elaborately in Sections 3–6.

2.4.1 Interest Flooding Attack

In an interest flooding attack, the attacker tries to deplete the NDN resources like PIT, network bandwidth, and producers' resources, by sending a large number of interest packets. This attack consumes NDN resources making them unavailable for legitimate consumers.

2.4.2 Cache Pollution Attack

In a cache pollution attack, the attacker tries to cache unpopular content in the NDN router by requesting unpopular data packets. Due to this attack the hit ratio of CS of NDN router decreases. Thus the request from a legitimate consumer has a low chance to get a cache hit.

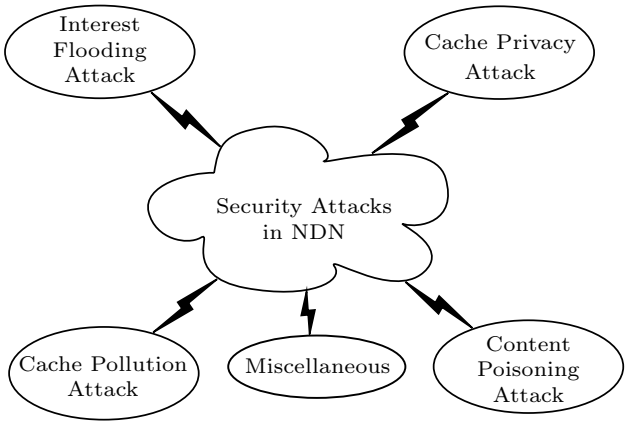

Fig.2. Security attacks in NDN.

2.4.3 Cache Privacy Attack

In a cache privacy attack, the attacker tries to find out whether the privacy-sensitive content is recently accessed or not. Privacy-sensitive contents are contents which can be associated with a user or a group of users. A recently-accessed item lies in the cache of the routers and is quickly responded to the requester. The attacker compiles a list of privacy-sensitive contents and requests them one by one to know whether they are cached or not by observing the delay in the content retrieval. If the content is found, then the attacker can infer that an intended user or a group of users have recently accessed that content. This attack enables the attacker to know the access pattern of the user, the type of content accessed, and other privacy-sensitive information contained in the content. Suppose, in an organization, there are few Spanish people. An attacker can create a list of popular contents which are accessed by most of the Spanish people and request for these contents. This will enable the attacker to find out the access pattern, likes and dislikes, of the Spanish people working in the organization.

2.4.4 Content Poisoning Attack

In the content poisoning attack, the malicious router sends a reply for the request with fake or corrupted content. These contents get stored in CS of other routers that are involved in communication. The contents further spread as other legitimate consumers request for these poisoned contents.

Table 2 shows attack types in NDN with the attacker, victim, affected security goals, and affected NDN data structure. In an interest flooding attack, the attackers are consumer applications that request non-existing interest packets and victims can be legitimate consumers, routers or producers. In the cache privacy attack and the cache pollution attack, the attacker and the victim both are consumer applications. In the cache privacy attack, the attacker breaches the confidentiality of the victim by finding recent cached items in CS. The cache pollution attack decreases the hit ratio of CS, thus degrading average delay at the consumer side. The content poisoning attack can be performed either by the router or by the producer. It affects consumer applications and routers.

3 Interest Flooding Attack

The main purpose of the flooding attack is to consume the network resources and make them unavailable for the legitimate consumer. This is done by sending a large number of interest packets. These interest packets create PIT entries in all the NDN routers from source to destination. These entries remain in PIT for a longer duration, thereby making PIT unavailable for the legitimate consumers. Gasti *et al.*[17] proposed three types of interest flooding attacks that are based on interest packets used for performing interest flooding attack: type-1 (existing or static), type-2 (dynamically-generated), and type-3 (non-existent). In type-1, the attacker continuously sends the interest packets for a set of existing contents. Due to in-network caching the nearby NDN routers cache these contents. Therefore, the repeated requests for the same contents are satisfied by these routers. This attack creates fewer PIT entries; therefore, this attack is ineffective in NDN.

In type-2, the interest flooding attack is performed by requesting for contents that are dynamically generated. As the contents are generated dynamically, it is not satisfied by the CS of NDN routers. The requests for these contents create entries in the PIT of NDN routers from source to destination. The newly created entries remain in PIT until the data packet corresponding to these PIT entries is received by the routers. The producer generates dynamic content for these interest packets. Under an attack scenario, the resources of the producer are wasted in satisfying such requests. Thus this attack affects consumers, routers, and producers.

In type-3, interest flooding attack is performed by requesting non-existing contents. The interest packets corresponding to non-existing content create PIT entries which are not satisfied. These entries remain in PIT till timeout, thereby making PIT unavailable for the legitimate consumer. This attack is more severe than the type-2 attack as these PIT entries which are created due to the attack remains in PIT till timeout whose duration is generally large.

Dai *et al.*[27] classified type-3 attacks into two categories as follows: 1) attacks using pre-existing prefix and 2) attack using random string. In attacks using pre-existing prefix, the interest flooding attack is performed by requesting dynamically created interest packet using the prefix of the existing producer. This prefix can be easily generated by appending random string, say "*/rand*", to existing prefix, for example, "*/UCLA*." These interest packets go to a specific producer (UCLA) and create entries in PITs. These PIT entries remain till timeout happens. Therefore, interest packets sent by a legal consumer do not get space in PITs. Fig.3 shows a demonstration of attacks using a pre-existing prefix. The attacker creates interest packets by concatenating a random string to a publisher's namespace "*/prefix*". These interest packets create entries in all the intermediate routers lying along the path from the attacker to the publisher. These entries remain in the PIT until timeout; therefore, consumer requests do not have space in the PIT.

In attacks using random strings, the interest flooding attack is performed using a random string as an interest packet. In old NDN specification[8], when there is no FIB entry found for an interest packet, the router forwards it from all the outgoing interfaces. These inte-

Table 2. Attack Characteristics and Security Goals Affected by Attacks

Attack Type	Attacker	Victim	Affected Security Goals	Associated Data Structure
Interest flooding attack	Consumer application	Consumer application, router, & producer application	Availability	PIT
Cache privacy attack	Consumer application	Consumer application	Confidentiality	CS
Cache pollution attack	Consumer application	Consumer application	Availability	CS
Content poisoning attack	Router or producer application	Consumer application & router	Integrity & availability	CS

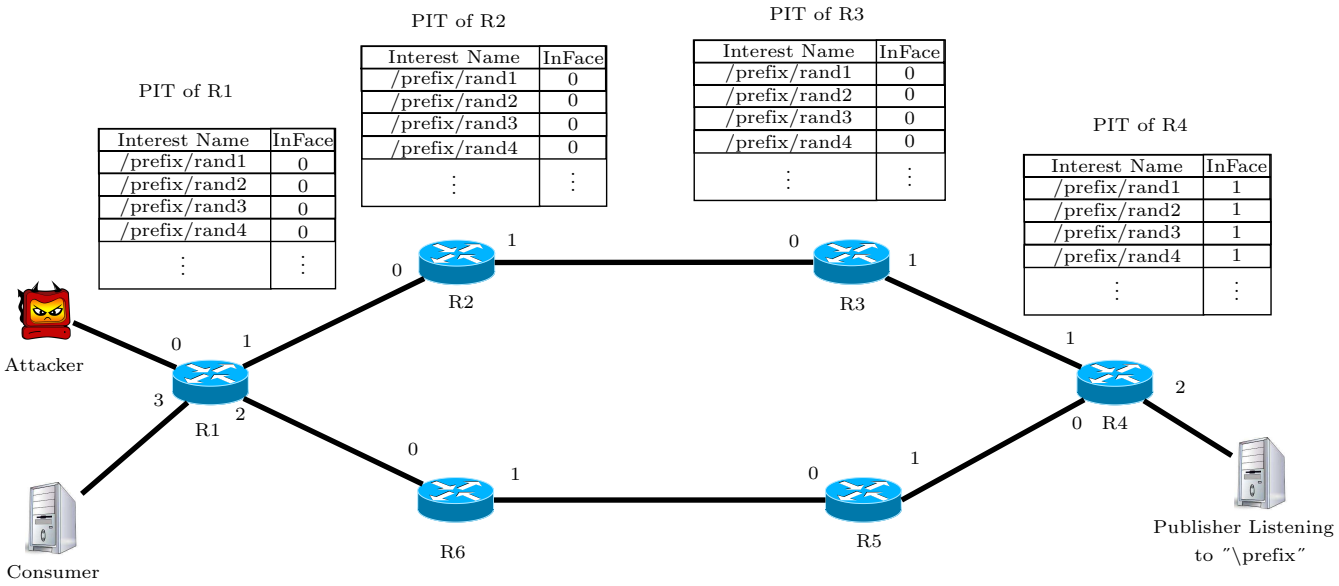

Fig.3. Interest flooding attack using pre-existing prefix. R: router.

rest packets create entries in PITs of in-between NDN routers. These entries remain in the PIT till timeout. However, current NDN architecture is resilient to this attack. If no entry is found in FIB for an interest packet then either packet is dropped, or a negative acknowledgment is sent.

Signorello *et al.*[28] proposed two new types of interest flooding attacks — blended interest flooding attack (bIFA) and chameleonic interest flooding attack (cIFA). In bIFA, the attacker uses existing and non-existing interest packets for performing the attack. This attack confuses the router by affecting detection matrices used by the router. In cIFA, the attacker requests non-existing interest packets having different prefixes. The attacker changes target prefix so that prefix-based countermeasures could not be able to mitigate the attack. Salah and Strufe[29] proposed a new type of the interest flooding attack called the collusive interest flooding attack. In this attack, the requests of attackers are satisfied by the collusive server just before the timer expires. Also, PIT entry does not expire but remains in the PIT for a longer duration. Therefore, timeout-based countermeasures do not work for this type of attacks. In the prefix hijacking attack[17,30], the attacker is a producer which announces the victim producer's prefix and drops all the interest packets. These interest packets create PIT entries and remain there until timeout. Table 3 shows the types of the interest flooding attack, affected entities, and their severity levels.

Based on the above discussion, the interest flooding attack can be divided into three categories as follows: 1) the interest flooding attack using existing content, 2) the interest flooding attack using the non-existing content, and 3) the interest flooding attack using both types of content (blended interest flooding attack). In the first category of interest flooding attack, the interest packet sent by an attacker can be replied by a producer via data packet.

Table 3. Attack Types and Their Effect on Network Entities

Attack Type	Affected Entity			Severity
	CA	PA	Router	
Static content[17]	✓	✗	✗	Low
Dynamic content[17]	✓	✓	✓	Low
Collusive interest flooding attack[29]	✓	✗	✓	High
Prefix hijacking[17]	✓	✗	✓	High
Pre-existing prefix[27]	✓	✗	✓	High
Random prefix[27]	✓	✗	✓	*
cIFA[28]	✓	✗	✓	High
bIFA[28]	✓	✗	✓	High

Note: CA: consumer application, PA: producer application, and *: not possible in current NDN.

The interest flooding attack using static content, the interest flooding attack using dynamic content, the collusive interest flooding attack, and prefix hijacking belong to this category. In the second category of interest flooding attack, the publisher does not send any data packet for the attacker's interest packet. The interest flooding attack using a pre-existing prefix, interest flooding attack using a random prefix, and cIFA belong to this category.

In the third category of interest flooding attacks, the interest packets sent by the attacker are a mixture of interest corresponding to the existing and the non-existing content. Fig.4 shows the types of interest flooding attacks in NDN.

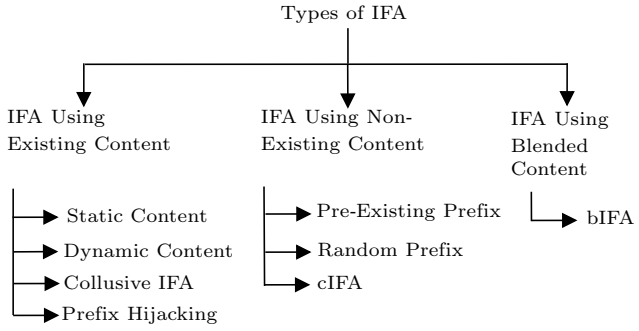

Fig.4. Types of interest flooding attacks.

3.1 Detection of Interest Flooding Attack

The detection approaches can be classified into three major categories, i.e., statistical modeling, probabilistic modeling, and miscellaneous approaches. Most of the authors[17,18,27,29,31–34] discussed the statistical threshold-based approaches. Wang *et al.*[35] and Nguyen *et al.*[36] gave a Fuzzy-based and Hypothesis Testing statistical approach. In probabilistic modeling, authors used Entropy-based[37,38], and Markov-based approach[39]. Some of the authors have proposed approaches based on Wavelet Analysis[40], Machine Learning[41–43], and Micropayment[42]. The classification of interest flooding attack detection approaches is shown in Fig.5. More details of detection approaches are discussed later along with countermeasures.

One more important characteristic of any detection approach is its granularity. The granularity of detection determines how fine-grained the detection is. In the interest flooding attack, detection granularity can be an interface, a namespace, the number of PIT entries or any combination of these three. Having a fine-grained detection helps in the proper mitigation of the attack. However, fine-grained detection incurs extra overhead to the router. Therefore, a fine-grained detection with low computational overhead is preferable.

3.2 Mitigation of Interest Flooding Attack

Mitigation of interest flooding attack can be done via two approaches: push-back approach and trace-back approach. In the push-back approach, the router, which detects the attack, informs the immediate downstream router about it by sending a message through a data packet or interest packet. The data packet is preferred over the interest packet for sending a message as the receiving router facing the interest flooding attack may drop the interest packet. After receiving this packet, the router applies a filter to the malicious interface, which may be interface-based or namespace-based. The attack is pushed back towards the attacker; hence the approach is named as push-back.

In the trace-back approach, the router which detects the attack sends an alert message to the gateway router through which the attack traffic is originated. The gateway router then applies a namespace-based filter or an interface-based filter to the malicious interface. The approach traces the gateway router, which originates the attack; hence it is named as a trace-back approach.

Fig.6 shows the categorization of interest flooding attack mitigation approaches. Most of the existing mitigation approaches are either push-back approaches or trace-back approaches. There are also three other approaches, namely, disabling PIT exhaustion, interest cash, and cryptographic route token, which are less frequently used for the mitigation of some specific interest flooding attacks. These approaches may be collectively classified as miscellaneous approaches and have been discussed in Subsection 3.3.

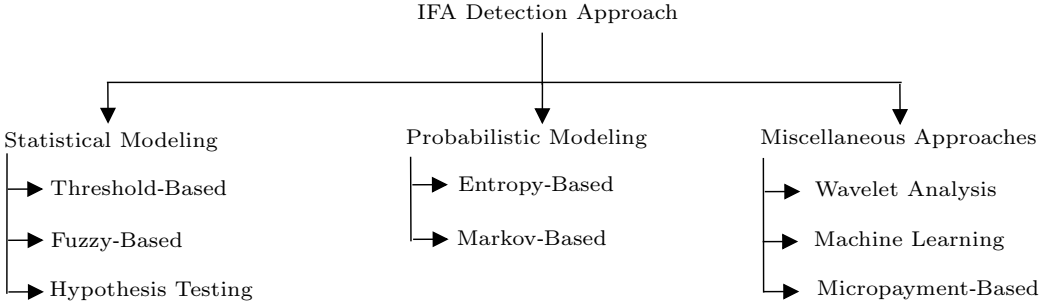

Fig.5. Types of interest flooding attack detection approaches.

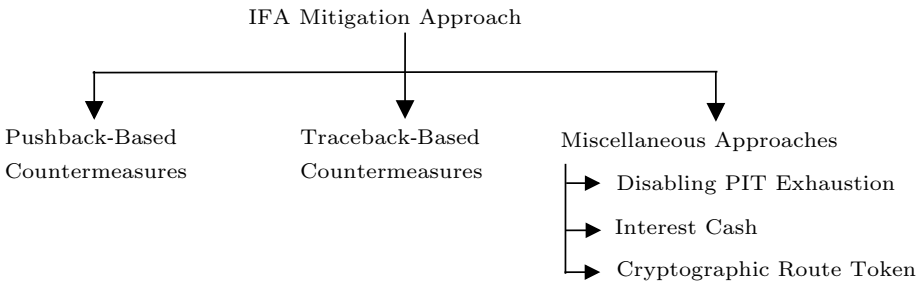

Fig.6. Classification of interest flooding attack mitigation approaches.

3.3 Tackling Interest Flooding Attack

This subsection presents the key research done to handle the interest flooding attack in chronological order. In 2013, Gasti *et al.*[17] introduced the NDN-specific DoS attack as interest flooding. Three types of interest flooding attack as mentioned in Section 3 have been proposed. The authors also proposed a tentative countermeasure to detect the interest flooding attack. The proposed countermeasure triggers a push-back mechanism that reduces the PIT quota of the malicious namespace for a given interface. The article by Gasti *et al.*[17] gives the direction and motivation to the researchers of this field, but it lacks the implementation and evaluation of the proposed detection and mitigation approaches.

Afanasyev *et al.*[18] proposed three algorithms as follows: 1) token bucket with per interface fairness, 2) satisfaction-based interest acceptance, and 3) satisfaction-based push-back algorithm. All three algorithms are based on applying a limit to the number of forwarded interest packets for each interface. The first algorithm is similar to the classical token bucket algorithm, with the difference that the tokens available for each outgoing interface are equally divided among all the incoming interfaces. In the second algorithm, each interface learns from the past unsatisfied interest packets. The router maintains the satisfaction ratio for each interface and allows interest packets based on this satisfaction ratio. The interest packet will be dropped or not depends on a random variable that is a function of satisfaction ratio. In the third algorithm, each incoming interface has its limit to satisfy the interest packet. Routers announce these limits to downstream neighbors, according to which neighbors set their limit. This approach helps legitimate interfaces to improve their statistics. Satisfaction-based push-back is better than the other two techniques in mitigating interest flooding attacks. These approaches apply a probabilistic filter on the malicious interface, which may also let the legitimate packets suffer.

Dai *et al.*[27] used the PIT size as a metric to detect the interest flooding attack. When it goes beyond a predefined threshold, then the router applies a traceback mechanism. The router looks for the unsatisfied interest packet having the longest name and sends a spoofed data packet as a reply to that interest packet. This packet reaches the originating router which applies a filter to the interface from where the request originates. This approach takes only the PIT size as the parameter; therefore, a temporary burst of traffic from a legitimate user can also trigger detection approach, which may result in the false detection. It can be inferred that Afanasyev *et al.*'s approach[18] is better in terms of the detection accuracy.

Compagno *et al.*[31] proposed a framework for detection and mitigation of interest flooding attacks, called Poseidon. Poseidon uses two parameters for attack detection as follows: 1) the ratio of incoming interest packets and outgoing data packets and 2) PIT space per interface. When both the parameters go beyond a predefined threshold, then the attack is detected on the interface. After detecting the attack, the interest packet is dropped, and the router sends an alert message from the detected interface. This alert message contains the timestamp of message generation and reduced rate. A router receiving this alert message decreases the threshold value of the above parameters. This approach is better than approaches in [18, 27] for the detection of the malicious interface as it uses satisfaction ratio, and the PIT size. A small burst of legal interest packets can also result in a temporary decrement of satisfaction ratio. Therefore, the detection using satisfaction ratio alone may result in a false detection. This approach applies mitigation on the interface. Thus, incoming legal interest packets may also suffer from interest flooding attacks.

Tang *et al.*[32] proposed a detection approach hav-

ing two phases, i.e., rough detection phase and accurate identification phase. In the rough detection phase, attacking interface is detected based on the relative strength index (RSI) of each interface. RSI for an interface is defined in (1).

$$RSI = \frac{\widehat{I_n}}{\widehat{I_n} + \widehat{D_n}} \times 100\%. \quad (1)$$

Here $\widehat{I_n}$ is the average number of incoming interest packets, and $\widehat{D}_n$ is outgoing data packets of an interface at the n-th time. They use the exponentially weighted moving average (EWMA) with a α coefficient to calculate the incoming interest packets and the outgoing data packets periodically. The value of $\widehat{I_n}$ and $\widehat{D}_n$ can be calculated by (2) and (3) respectively.

$$\widehat{I_n} = \alpha \times \widehat{I_{n-1}} + (1-\alpha) \times I_n, \quad (2)$$

$$\widehat{D}_n = \alpha \times \widehat{D_{n-1}} + (1-\alpha) \times D_n. \quad (3)$$

Here I_n and D_n are the total number of incoming interest packets and outgoing data packets received on an interface in the n-th period respectively. The router observes the value of RSI. When it becomes higher than a predefined threshold, then the router reports this interface to the accurate identification phase.

In the accurate identification phase, the router counts all the interest packets carrying the largest expired prefix at the reported interface and calculates the expired ratio. If the expired ratio of a named prefix exceeds a predefined threshold, then the detected prefix is the abnormal prefix. This approach is more fine-grained than Compagno *et al.*'s approach[31] as it detects malicious interface as well as the malicious prefix. However, in terms of malicious interface detection, Compagno *et al.*'s approach[31] is better than Tang *et al.*'s approach[32] as it takes satisfaction ratio as well as PIT size for the detection. Also, the authors of [32] did not mention any countermeasure after detecting malicious prefix.

Wang *et al.*[33] proposed an approach called Disabling PIT Exhaustion (DPE) to counter interest flooding attack. In this approach, each router has a malicious list called m-list. This list records the number of expired interest packets per namespace. When this value exceeds a predefined threshold, then the namespace is considered malicious. The interest packets corresponding to the namespace are marked and not recorded in the PIT, but forwarded further. Therefore, malicious interest packets do not affect routers as these interest packets are not stored in PITs. These malicious namespaces remain in m-list until decay time. This approach takes expired interest packets per namespace as a parameter; therefore, cIFA can be performed on this approach. After the detection, the legal interest packets belonging to the malicious namespaces also suffer from this approach. This scheme does not stop the attacker; it only lessens the effect of interest flooding attack.

Wang *et al.*[35] applied a fuzzy-based filter on every NDN router to detect interest flooding attack. In the detection phase, the router periodically monitors the PIT occupancy rate (POR) and PIT expiration rate (PER). POR is the ratio of the number of current PIT entries and the maximum PIT entries in a given time interval. PER is the ratio of the number of expired PIT entries and the total number of pending PIT entries in a given time interval. Both POR and PER act as fuzzy variables which are used by the detection mechanism for the detection of interest flooding attack. The attack is detected when one of the fuzzy variables becomes high. After detecting the attack, the prefix having too many PIT entries or expired PIT entries can be considered as a malicious prefix. The router sends a push-back message containing the information of malicious prefix to notify other routers about the attack. Any router, which receives this push-back message, drops all the interest packets that belong to the malicious prefix. In this paper, the detection is based on POR which increases when the rate of interest packets arrival increases. However, sometimes POR can also increase due to the legal burst of traffic which may lead to a false positive result. Therefore, Compagno *et al.*'s approach[31] is better than this approach. Also, this approach is ineffective against collusive interest flooding attack since it is based on PIT expiration rate.

Li and Bi[44] used an application-based approach to counter interest flooding attacks called Interest Cash. In Interest Cash, the consumer requests a meta-puzzle from the publisher before requesting content. The meta-puzzle (for example, "/prefix/puzzle") consists of a timestamp t, a random string x, a solution length l (bits), a puzzle difficulty factor k ($l > k$) and an expiration time T. The consumer has to generate the final puzzle for getting actual content. The final puzzle for the consumer is to find the valid cash for interest name n (for example, "prefix/content_1/") according to Definition 1.

Definition 1 (Valid Cash). *For a meta-puzzle $(t; x; l; k; T)$, the valid cash for the content with name n is an l-bit string s, which makes the first k bits of string hash $(n + t + x + s)$ all 0.*

After calculating the valid cash s, the content consumer should append s and t to the name n and send the interest with the reconstructed name (for example, "/prefix/content_1/t/cash_1") to the content provider. The content consumer can save this meta-puzzle to request other contents. After the expiration time T, the consumer has to solve a new puzzle to get new content from the publisher. Interest Cash makes it difficult for the attacker to generate a valid request to the publisher. Authors of [44] showed that an adversary needs at least 300 times the computation resources of the content provider to commit a successful attack. Although this approach makes it difficult for the attacker to perform attacks, solving puzzle after each time out is a costly operation.

Karami and Guerrero-Zapata[41] proposed a multi-objective RBF-PSO based approach for the detection of interest flooding attack. In this approach, 12 features are used for the detection of interest flooding attacks. The authors detected interest flooding attacks using the pre-existing prefix and the random prefix along with prefix hijacking attack. They used NSGA-II for choosing optimal centers for RBF network which is based on optimizing mean square error (MSE) and Davies-Boulding index (DBI). After finding centers, the widths of the RBF units and output weights are optimized using PSO. This trained network is used for the detection of interest flooding attack. After detection, the router sends an alert message to all the interfaces which are detected as malicious. The alert message contains the time stamp corresponding to the generation time of the alert message, the new reduced rate, and the wait time of reduction period. After receiving this alert message, the router will reduce forwarding limits of its interfaces according to the reduced rate present in the alert message.

The authors of [41] have not analyzed the features which are necessary for attack detection. The detection is based on a pre-trained neural network which is trained using the traffic of fixed topology. Therefore, this approach highly depends on topologies, i.e., the same trained classifier cannot be used for different topologies.

Nguyen *et al.*[36] proposed an approach for the detection of interest flooding attack based on statistical hypothesis testing. They gave a parametric model for the packet-loss rate in interest flooding attacks. This model is used to design the generalized likelihood ratio test (GLRT) for a scenario where the packet-loss rate is unknown. This model takes the desired false alarm probability as a parameter and calculates the detection threshold. Authors[36] did not give any countermeasure approach after detection.

Vassilakis *et al.*[34] proposed a countermeasure against interest flooding attack in which the edge router detects a malicious user (u) by tracking the number of expired PIT entries per time unit ($N_{\exp}(u)$). This approach consists of three phases as follows: 1) attack detection phase, 2) rate reduction and blocking phase, and 3) attack notification phase. In the first phase, the users are classified into three categories, i.e., legitimate (L), suspicious (S) (possible attackers), and malicious (M) based on two thresholds (T_{low} and T_{high}). If $N_{\exp}(u)$ is lower than T_{low}, then the user is legitimate. If it lies in between T_{low} and T_{high}, then the user is suspicious. Else, $N_{\exp}(u)$ is greater than T_{high}, which means the user is malicious. In the second phase, the data rate of the users is assigned according to (4).

$$R_{\text{new}}(u) = \begin{cases} R_{\text{old}}(u), & \text{if } u \in L, \\ \dfrac{\alpha R_{\text{old}}(u)}{T_{\text{high}} - T_{\text{low}}}, & \text{if } u \in S, \\ 0, & \text{if } u \in M. \end{cases} \tag{4}$$

Here $R_{\text{new}}(u)$ and $R_{\text{old}}(u)$ are the new and the old rate of the user u respectively. In the third phase, a notification message is sent to other routers to inform an ongoing attack.

Alston and Refaei[45] proposed an approach in which the interest packet carries routing information. This approach uses a special data structure called cryptographic route token (CRT) for holding route information. Initially, a consumer constructs an interest packet having empty CRT and forwards it to the next hop router. On receiving this interest packet, the router checks the validity of the received interest packet and adds the routing information to the CRT. This interest packet subsequently reaches the producer where the producer constructs the data packet and forwards it. The router receiving the data packet removes its information from CRT and forwards it to the next router. In this way, this data packet is received by the consumer. The confidentiality and the integrity of this CRT are assured by using asymmetric cryptography. This approach eliminates the reliability on the PIT. The performance of this approach can be improved by using this approach when the PIT size goes beyond a certain limit. The main limitation of this approach is the computational cost due to the reconstruction of the interest packet at each router on the path to the producer.

Xin *et al.*[37] proposed an approach based on non-parametric cumulative SUM (CUSUM) to detect the change in the mean value of entropy of names of received interest packets on an interface. If the entropy of the interface exceeds a predefined threshold, then the attack is detected. After the detection, the malicious prefix is identified using the relative entropy of the current sample, and the sample observed a little earlier. Mitigation is applied by sending spoofed data packet towards the gateway router which applies a limit on the incoming malicious interest packets on the detected interface. This approach is based on observing the change in entropy based on the name of interest packets; therefore, this approach cannot detect cIFA.

Ding *et al.*[39] proposed an approach based on three interest flooding attack properties, i.e., distributed attackers, the high rate of requests, and requesting non-existential interest packets. These properties are quantified as POR, PER, and potential attacker ratio for each prefix. Based on these three property variables, the authors developed a Markov model for the state of a router. These states are "Normal", "Risk", and "Unknown". These states are calculated using α, where α is the square root of the sum of the square of each property variable mentioned above. The router forwards the interest packet along with the state information. Each router maintains a retransmission cash. The retransmission cash has the names of interest packets that are not satisfied. When a PIT entry timeout then the name of the corresponding interest packet is inserted in to retransmission cash. If this interest packet is requested one more time, then this interest packet is forwarded further having state information "Normal" and the name is removed from retransmission cash. Otherwise, the router sends an interest packet with calculated state information. On receiving an interest packet, the router matches the state information of the interest packet with the expected state information. If both the states are "Risk" then the router discards all the interest packets belonging to the malicious prefix until the PIT size of the server drops to a safety value. This approach is better than that in [35] because besides POR and PER it also takes potential attacker ratio as a parameter. The detection is based on the namespace of the received interest packets; therefore, cIFA can be performed on this approach.

Salah *et al.*[46,47] proposed a framework to detect and countermeasure interest flooding attacks known as Coordination with Lightweight Monitoring (CoMon). This framework consists of three components, i.e., domain controller (DC), NDN routers (NRs), and monitoring routers (MRs). Each domain has a DC which monitors MR. NRs are normal NDN routers with a capability to be informed about the mitigation of interest flooding attack. MRs are special NDN routers which monitor PIT utilization ratio (PUR) and PIT expiration rate (PER). PUR is the ratio of the maximum number of PIT entries to the PIT space observed in a fixed time interval. PER is the ratio of the number of expired PIT entries to the sum of the number of expired and satisfied PIT entries. When PUR exceeds a predefined threshold, and the value of PER is positive, then the attack is detected. After the detection, expired named prefixes are considered as malicious by MRs. The MRs calculate PER for each malicious named prefix and send this information to DC which aggregates this information and finds out infected named prefixes. DC sends this information to MR which calculates the PER for each infected name-prefix. This PER is used for probabilistic rejecting of incoming interest packets. Salah *et al.*[46] evaluated their scheme on ndnSIM simulator using AS-3967 topology. The satisfaction ratio for legitimate interest packets under interest flooding attacks was 12% and after applying defense mechanism this ratio increased by 78%. Salah *et al.* extended their work in [29] to work against collusive interest flooding attacks. This approach uses its own infrastructure over NDN; therefore, it incurs more message overhead. Signorello *et al.*[28] showed that this approach is ineffective against cIFA and bIFA.

Xin *et al.*[40] proposed a detection approach for collusive interest flooding attacks based on wavelet analysis[48]. The authors showed that the power spectral density (frequency distribution of traffic for the namespace) of traffic corresponding to collusive interest flooding attacks is low compared with other traffic. The authors used the average attack strength of all the subbands for the detection of the attack. The threshold for detection is chosen experimentally. This approach is only applicable to collusive interest flooding attacks while other types of interest flooding attacks can be possible in this approach. Authors of [40] did not give any mitigation approach after the detection of the attack.

Kumar *et al.*[42] implemented interest flooding attacks using CCNx-based implementation and ndnSIM-based simulation for a linear topology. Six parameters were chosen for the detection of interest flooding attack using six machine learning-based approaches. The accuracy of the interest flooding attack detection for

the simulation and implementation is found almost the same.

Wang *et al.*[49] gave an approach to counter interest flooding attacks by employing micropayments on NDN nodes (consumers and routers). Initially, each node has some virtual money (VM) which is managed by NDN trusted authorities. NDN nodes use this VM for forwarding NDN packets. The misbehavior of nodes can be observed based on their access use of VM.

Zhi *et al.*[38] gave an interest flooding attack detection approach based on computing Gini impurity (GI)[50]. In this approach, the router computes the GI of the set of namespaces observed in a given period. The attack is detected when the relative increase in GI goes beyond a predefined threshold. After detecting the attack, the malicious prefix is detected by replacing the probability of each namespace in the old set with the probability of the same namespace in the new set. The router calculates the difference between these two sets. If this difference is negative, then the prefix is regarded as malicious. Mitigation is done by limiting the forwarding rate of the malicious interest packets. The router also sends a notification to downstream routers. The detection is based on the namespace of the received interest packets. Therefore, cIFA can be performed on this approach because in cIFA the attacker attacks on more than one target prefix that will make GI of namespaces remain the same.

Zhao *et al.*[51] proposed a new sophisticated interest flooding attack. A group of attackers perform this interest flooding attack by sending non-existing interest packets with a slow rate at the beginning and then slowly increase the rate. This attack has a greater effect on intermediate routers near the attacker than on the access routers.

Kumar *et al.*[43] applied ranking based on information gain for 12 different features. Out of 12 features, nine features were found significant for the detection of interest flooding attack.

3.4 Findings

Table 4 summarizes interest flooding attack types, and the detection and mitigation approaches. The interest flooding attack using pre-existing prefix is the most addressed problem. The other types of interest flooding attacks are less critical. Most of the detection and mitigation fails to mitigate cIFA and bIFA. Hence, it is a challenge for future researchers. A good detection approach should have fine-grained detection, and less computation and storage overhead. Most of the countermeasures use either push-back or trace-back mechanism. For reducing the effect of the interest flooding attack, the initial push-back mechanism is necessary. The trace-back helps to stop the attack being performed by the attacker. A combination of push-back and trace-back is more effective than push-back or trace-back alone. The efficiency of mitigation depends on the granularity of detection, i.e., if the detection is more fine-grained, then the mitigation can be applied to stop only specific namespace on a specific interface. The implementation of interest flooding attack and its detection approach and mitigation is done mostly on ns-3 based ndnSIM[52] simulator.

4 Cache Privacy Attack

Router side content caching is the fundamental feature of NDN, which reduces the network congestion, optimizes bandwidth utilization, and provides fast access to the content. However, an attacker can access these cached contents and find whether these contents are recently accessed by someone else or not. After knowing the time of access and content name, an attacker can associate this content to a user, thus breaching the privacy of the user. This attack is known as cache privacy attack. The cache privacy attack does not hinder communication, but the attacker can use the information gained from this attack for his/her benefit. It is challenging to detect this attack as the network does not get affected by this attack.

Protecting all the cached contents from the attacker is a big overhead for the router. Also, it is not necessary that all the cached contents are privacy-sensitive. Therefore, securing only the privacy-sensitive contents could reduce the overhead. Deciding whether content will be private or public is another issue. Privacy depends on whether the content can be associated with a particular context (political view, religious beliefs, etc.), a user and a location at a given time. Determining which data will be private and how the privacy technique will be implemented is also a crucial issue for NDN. The producer can make the content as private by either declaring predefined namespaces as private or setting the extra privacy bit of the content itself. The first way is too complicated as the router has to know all the namespaces that are private and these namespaces must get updated. The second way is feasible because producers can control it. The producer can set the privacy bit before replying to the data packet. After deciding how privacy will be implemented, the main

Table 4. Interest Flooding Attack Types and Detection & Mitigation Approaches

Reference	Attack Type	Detection			Countermeasure	
		Type	Parameters	Granularity	Type	Topology
Gasti *et al.*[17], 2013	Interest flooding attack for static, dynamic & non-existing content & prefix hijacking	Threshold-based	Pending interest per outgoing interface, incoming interest per interface & pending interest per namespace	Interface & namespace	Push-back	
Afanasyev *et al.*[18], 2013	Interest flooding attack using pre-existing & random prefix	Threshold-based	Outgoing interest (1st Algo.), forwarded & unsatisfied interest (2nd & 3rd Algo.)		Push-back	Tree & AT & T
Compagno *et al.*[31], 2013	Interest flooding attack using pre-existing & random prefix	Threshold-based	Ratio of incoming interest & outgoing data & PIT size	Interface	Push-back	AT & T
Dai *et al.*[27], 2013	Interest flooding attack using random prefix	Threshold-based	PIT size	Interface	Trace-back	EBONE (AS1755)
Tang *et al.*[32], 2013	Interest flooding attack using pre-existing & random prefix	Threshold-based	RSI (1st phase) & largest expired prefix (2nd phase)	Interface & namespace		Self-made
Wang *et al.*[33], 2013	Interest flooding attack using pre-existing & random prefix	Threshold-based	Expired interest per namespace	Namespace	Disabling PIT exhaustion	Linear and AT & T
Wang *et al.*[35], 2014	Interest flooding attack using pre-existing & random prefix	Fuzzy-based	POR & PER	PIT entries	Push-back	AT & T
Li and Bi[44], 2014	All types				Interest cash	Self-made
Karami and Guerrero-Zapata[41], 2015	Interest flooding attack using pre-existing & random prefix & prefix hijacking	Machine learning based	12 parameters	Interface	Push-back	DFN and AT & T
Nguyen *et al.*[36], 2015	Interest flooding attack using pre-existing prefix	Hypothesis testing	Incoming interest & outgoing data	Interface		Tree
Vassilakis *et al.*[34], 2015	Interest flooding attack using pre-existing & random prefix	Threshold-based	Expired PIT entries	PIT entries	Push-back	Self-made
Alston and Refaei[45], 2016	All types				Cryptographic route token	Internet like
Xin *et al.*[37], 2016	Interest flooding attack using pre-existing prefix	Entropy-based			Trace-back	Tree
Ding *et al.*[39], 2016	Interest flooding attack using pre-existing prefix	Markov-based	POR, PER & PAR	PIT entries	Push-back	Tree and AT & T
Salah and Strufe[29], 2016	Collusive interest flooding attack & interest flooding attack using pre-existing	Threshold-based	Satisfaction ratio per namespace per Inter-face	Namespace	Push-back	AS-3967 & AS-3257
Signorello *et al.*[28], 2017	cIFA & bIFA					AS-3967
Xin *et al.*[40], 2017	Collusive interest flooding attack	Wavelet analysis	Incoming interest per namespace	Namespace		China Telecom
Kumar *et al.*[42], 2017	Interest flooding attack using pre-existing & random prefix	Machine learning approaches	6 parameters	Router		Linear & DFN
Wang *et al.*[49], 2017	All types	Micropayment-based	Virtual money	Router or consumer		Tree & net like
Zhi *et al.*[38], 2018	Interest flooding attack using pre-existing prefix	Entropy-based	Incoming interest per namespace	Namespace	Push-back	Tree
Salah *et al.*[47], 2019	Interest flooding attack using pre-existing & random prefix	Machine learning approaches	6 parameters	Interface		Tree & DFN

challenge is to determine the entity (consumer or producer) which will decide whether the content is private or not.

If a consumer wants a private content, then it can inform the producer by setting an extra privacy bit in the interest packet. However, if only the consumer has control over deciding the privacy of the content, then two problems arise. First, if the consumer sets the privacy bit for all the interest packets before requesting them, then the router will apply countermeasure for all the data packets corresponding to these interest packets, which may lead to the degradation of NDN performance. Second, if a consumer requests a private content without setting the privacy bit, then the privacy of the consumer gets affected. Therefore, the privacy could not be decided by a consumer alone. However, the producer alone cannot decide the privacy of content because a content which may seem public to a producer can be private for a consumer. Thus, the best way is that both the consumer and the producer decide the privacy of the content and the router must penalize the attacker.

Beyond these issues, it is also important that if a content is private, then the content must be handled in such a way that attackers could not infer whether it is cached or not. A simple solution for the problem is not to cache any private content, but that can result in the degradation of NDN performance. A better solution for this attack is to make difficult for an attacker to link privacy-sensitive contents to a user or a group of users.

In 2010, Lauinger[53] introduced cache privacy attack for CCN. He proposed three types of cache privacy attack, i.e., request monitoring attack or timing-based attack (TBA), object discovery attack (ODA) and data flow cloning attack (DFCA).

In TBA, the attacker tries to find recent access to privacy-sensitive contents from the nearby CS. The attack is performed by compiling a list of privacy-sensitive contents related to one or more users. After compiling the list, the attacker finds the hit time of 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 been already cached. The time taken to receive the reply (from the time request for the content is done) is known as the hit time. After getting the hit time, the attacker requests for privacy-sensitive contents one by one from the list to check whether it is cached or not. If the cache-hit occurs then the attacker can interpret that the user to whom the attacker has linked the content while making the list of the privacy-sensitive content must have accessed the cache recently. Fig.7 shows a demonstration of TBA.

The author of [53] extended his work in [54] and proposed a specialized technique to perform the above attack in which an attacker performs the cache privacy

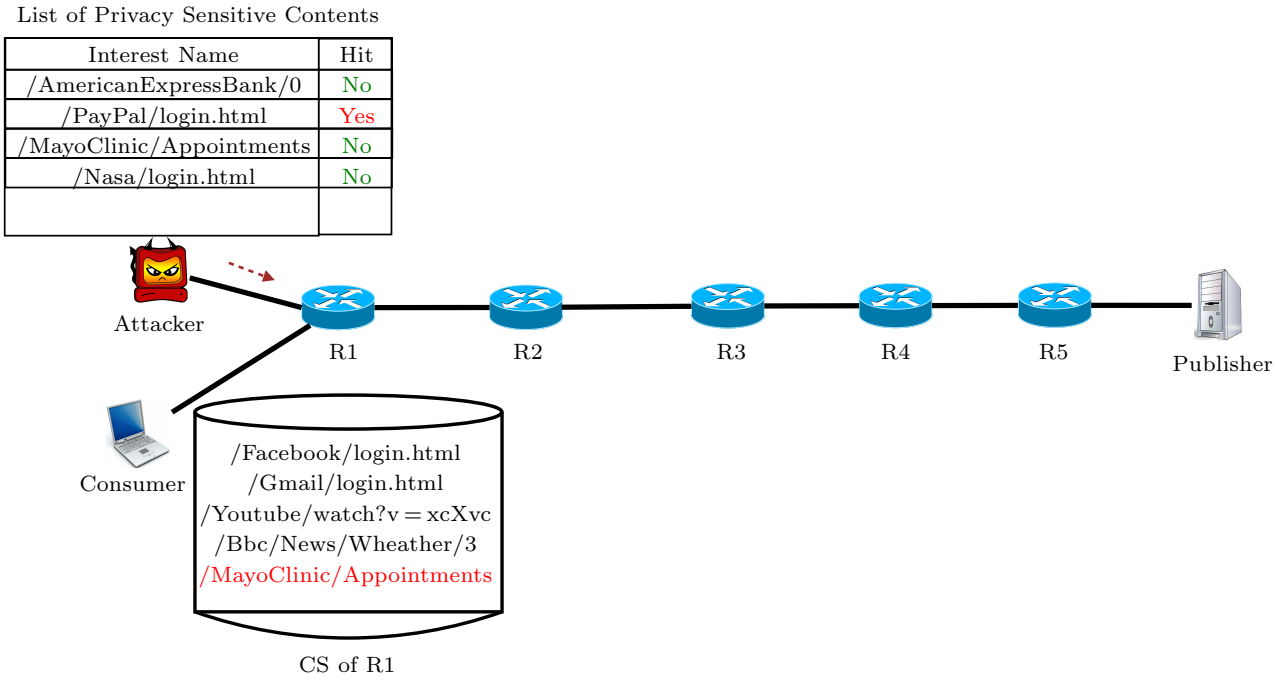

Fig.7. Timing-based attack.

attack by probing requests for a content C at a frequency f_p called probing frequency. The value of f_p must be greater than or equal to the reciprocal of t_c for ensuring the detection of C; here t_c is the average period for which the data packet remains in CS.

In ODA, an attacker initially sends the interest packet having the root namespace (represented as "/") in the name field of the interest packet. The router responds by sending any cached data packet. The attacker sends another interest packet after setting the exclude field to the name of the data packet received earlier. This time the attacker gets a new data packet as a reply. By repeating this action, an attacker can know which data is in CS.

In DCFA, the attacker replicates the on-going flow in the cache by inspecting flow-based messages in the cache. In this attack, the attacker knows the syntax and semantics of the conversation messages. The attacker composes this message for a specific consumer or a group of consumers and sends it to the router. If the same conversation is going on, then the router replies with the corresponding data packet. Now the attacker can request subsequent packets by composing the next message based on syntax and semantics. For example, the Voice-over-CCN message uses the naming scheme "*/domain/user/call-id/rtp/sequence-number*" for the voice data, exchanged during a call. The attacker can perform object discovery attack to know the on-going call instance such as "*/tid/john/1234/rtp/*". The attacker can predict future data packets by requesting in the same way as done by the original requester.

Arianfar *et al.*[19] proposed two types of cache privacy attacks that are performed by the government or a higher authority, i.e., name-watch list attack (NWLA) and content analysis attack (CAA). In NWLA, the adversary compiles a list of restricted namespaces after examining. Whenever a request for the restricted data packet is made, the adversary either blocks/filters the request or records the user who requested the data. In CAA, the attacker inspects the content requested by the user for finding the wrong keywords, spam, etc. The attack can block the users (who requested such a content) for a small interval of time.

From the above discussion, the cache privacy attack can be divided into two categories based on who performs the attack, i.e., the attack performed by a malicious consumer application (attacker) and the attack performed by ISP or a government. TBA, ODA, and DFCA lie in the first category whereas NWLA and CAA lie in the second category. Fig.8 shows the types of cache privacy attacks in NDN.

4.1 Mitigation of Cache Privacy Attack

Most of the mitigation approaches[20,54−65] are proposed for the timing-based attacks. Cache privacy attack mitigation approaches have been divided into five categories, i.e., delay-based approaches, caching-based approaches, disabling packet fields, detection-based approaches, and miscellaneous approaches. In delay-based approaches, the router applies an additional delay to the data packet before replying it. Applying delay to the first k requests and applying delay for time t are the examples of this approach. In the caching-based approach, the cache structure or caching algorithm is changed to resist the cache privacy attack. Probabilistic caching, collaborative caching, and caching based on betweenness centrality are the examples of this approach. In disabling packet fields approach, the fields (scope and exclude) which make cache privacy attacks easy to perform are disabled. In detection-based approaches, the router detects cache privacy attacks based on traffic characteristics and then applies a countermeasure. Tunneling and publishing encrypted chunks of a file are classified under miscellaneous approaches. The classification of mitigation approaches is shown in Fig.9.

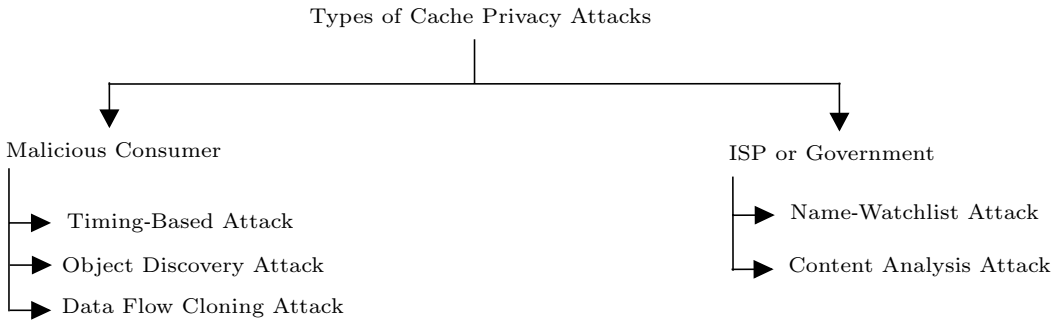

Fig.8. Types of cache privacy attacks.

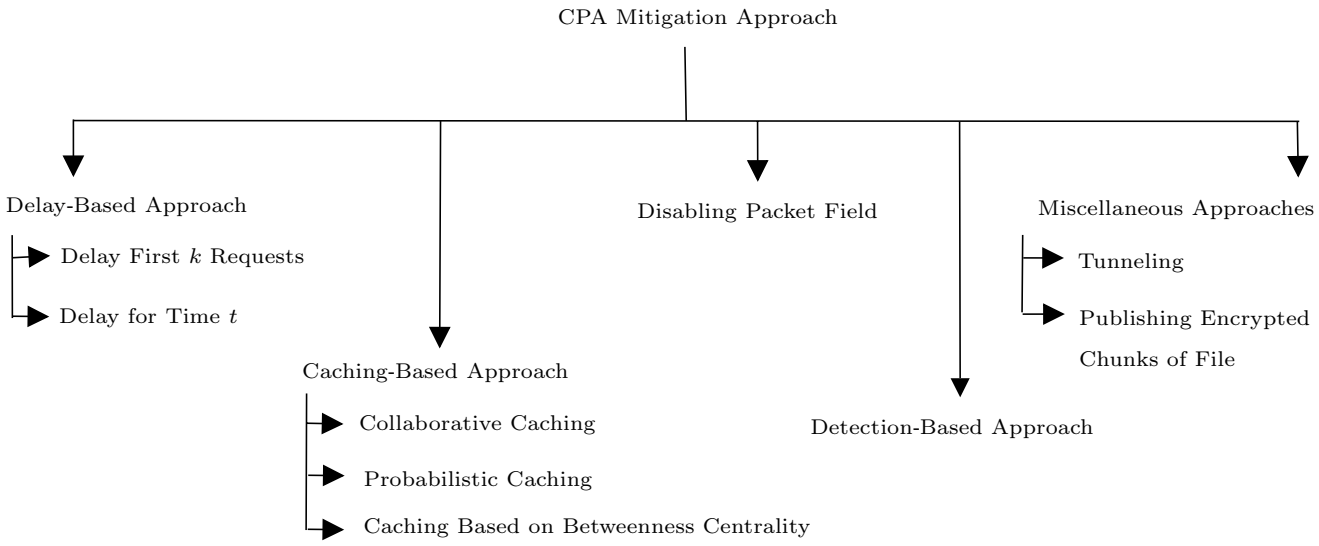

Fig.9. Cache privacy attack countermeasures.

4.1.1 Delay-Based Approach

In this approach, an additional delay is added before forwarding the data packet to the consumer or downstream router.

• *Applying Delay for First k Requests.* Acs *et al.*[55] proposed three approaches based on applying delay for the first k requests to a privacy-sensitive content. These approaches are Non Private Naive (NPN), Uniform Random Cache (URC), and Exponential Random Cache (ERC). In the NPN approach, k is fixed for all the contents in the cache. In the URC approach, a positive integer k is chosen in $(0, k]$ interval using uniform random distribution for each content c when it is cached. In the ERC approach, k is chosen using exponential distribution. Authors also calculated utility trade-off between privacy and caching. ERC shows 12% better performance than URC. They extended their work in [63] and gave a proof of their approach. They analyzed their approach for local and distributed adversaries. This approach mitigates cache privacy attacks by maintaining a trade-off between privacy and caching. Each router has to maintain the state of all the contents that are present in CS; therefore this approach incurs extra storage overhead to the router. Chaabane *et al.*[56] also mentioned the same countermeasure.

• *Applying Delay for Time t.* Mohaisen *et al.*[65] presented three approaches based on applying delay for time t in the case of a cache hit. These are vanilla approach, efficient approach, and low granularity approach. In the vanilla approach, the edge router maintains a per-user state, i.e., requested content's name and the times the consumer requests it. If the same consumer requests the same content, then no delay is applied. Otherwise, a delay is chosen which lies between the single hop distance and the RTT of the content. In the efficient approach, the router maintains a per-face state. Therefore, it incurs less storage overhead for the router. The low granularity approach takes the advantages of both approaches by maintaining per-user state in the access point and per-face states in the router. They have presented the evaluation and shortcomings of their approach in [58]. The vanilla approach and the low granularity approach could not be applied in the NDN due to the unavailability of User-ID like IP addresses. But, the efficient approach which maintains per-face states can be applied to the NDN architecture. This approach assures security as it is more resilient to cache privacy attacks than Acs *et al.*'s approach[55].

4.1.2 Caching-Based Approach

In the caching-based approach, the caching strategy is modified for preventing cache privacy attacks.

• *Probabilistic Caching.* Chaabane *et al.*[56] allowed the router to decide whether to cache content or not, based on a random function. It makes it difficult for an adversary to infer the state of a router's cache at a particular instant. There is no restriction for accessing content from the cache. Therefore, the attacker can still access the cached content. This approach only reduces the possibility of the attack.

• *Collaborative Caching.* Chaabane *et al.*[56] suggested to collaborate nearby caches and create a distributed cache, which leads to an increase in the cache

size for a user connected to nearby caches. When caches collaborate, the number of users connected to the cache also increases; therefore, the attacker faces difficulty in linking a particular content to a user. The authors did not provide any implementation details or the analysis of this approach. Maintaining a distributed cache is too expensive for the router. Kamath *et al.*[64] proposed an approach called Group Caching for Privacy in NDN (GCPiN). In GCPiN, the nearby access routers collaborate to form a group. Each router in the group has a GCPiN manager, which handles grouping of routers, distributed content caching, and securing privacy-sensitive content. This approach incurs extra overhead for maintaining groups of routers.

• *Caching Based on Betweenness Centrality.* Abani and Gerla[62] proposed a caching scheme based on betweenness centrality (BC). BC measures the times a specific node lies on the shortest path between all pairs of nodes in a network. The authors proposed that caching of privacy-sensitive content should be done on nodes having a high value of BC. This will make it difficult for an attacker to associate the content to a specific user. This approach is better than other caching-based approaches, but it causes more delay for privacy-sensitive contents as the privacy-sensitive contents get cached near the backbone routers.

4.1.3 Disabling Packet Field

Lauinger *et al.*[20] suggested disabling scope field and exclude field because they help an attacker to perform the attack. The attacker can set the value scope field to 2, thus allowing requests to propagate up to a single hop; this helps the attacker to perform attacks faster. The exclude field helps an attacker to perform ODA by first requesting root namespace ("/") for getting any cached data item and then sending a request message excluding the previously received data item using the exclude field. In this way, the attacker can know the cached data items in the CS. Disabling the scope or exclude field can result in the loss of NDN performance, as the scope field helps to restrict the further propagation of NDN interest packets and the exclude field helps the consumer to exclude the undesirable data packets. In the latest NDN packet format specification (0.3), the exclude field is removed from NDN. However, the hop-limit field exists in NDN, which is similar to the scope field.

4.1.4 Detection-Based Approach

Ntuli and Han[66] proposed an approach which detects cache privacy attack by analyzing attackers' behavioral parameters, i.e., the high-interest rate and the high cache rate. These parameters are used by the detection approach for calculating the trust value of the host connected to the gateway router. However, the cache privacy attack is more flexible than the interest flooding attack or cache pollution attack as the attacker only needs to request interest packets to perform this attack. Therefore, any fixed detection approach cannot entirely resist this attack.

Gao *et al.*[57] proposed an approach for detecting cache privacy attacks similar to that in [66]. This approach is based on detecting the high-interest rate, the high cache hit rate, and repeat requests for multiple contents in a short period. Gateway router calculates the credit score of all the users using the parameters mentioned above. If this credit score becomes less than a predefined threshold, then the attack is detected. After detecting the attack, the router replies the data packet to the attacker after adding a delay. This delay depends on the RTT of the content. Dogruluk *et al.*[60] also proposed a similar approach to [57, 66]. Kumar *et al.*[67] gave an approach to detect the pattern of the timing-based attack. If this pattern is detected for a threshold number of times, then a countermeasure is applied by delaying the data packet before replying.

4.1.5 Miscellaneous Approaches

There are two other approaches which have not been included in the previous categories.

• *Publishing Encrypted Chunks of File.* Arianfar *et al.*[19] proposed an approach for detecting NWLA and CAA. This approach is based on publishing encrypted chunks of a file. The publisher divides an original file into chunks of equal sizes and intermixes each piece to a piece of a cover file, resulting in blocks of files. When a user requests these files, the publisher replies with selective blocks of files. Before intermixing these chunks, the publisher applies a random function and a cover to each chunk of the file which is known to both the user and the publisher. On receiving these blocks, the user applies reverse function and integrate chunks to receive the original file. In this scheme, extra files are added to the chunks of the original file, thereby additional memory is used. Also, this scheme consumes CPU cycles for processing files at the publisher as well as at the consumer end.

• *Tunneling.* Lauinger *et al.*[20] suggested tunneling in which systems like Andana[68] can be used to tunnel the data packet directly to the client. Tunneling may lead to the loss of NDN performance, as in-network caching is not possible in tunneling. The authors suggested that it can be used only for privacy-sensitive content.

4.2 Findings

The cache privacy attack is a critical attack for ICN type networks as they use caching in the router itself. The attack does not influence network resources, and hence it is tough to detect. Table 5 shows different types of cache privacy attacks and the countermeasures. Most of the studies[20,54−58,60,62−65] have focused on TBA, which is more severe than other types of cache privacy attack. The detection, mitigation, and the evaluation of the cache privacy attack are done mostly using CCNx code base. Detection-based approaches[57,60,66] are in-effective as the pattern of the attacker can vary. Caching-based schemes like collaborative caching[64] use complex infrastructure which is extra overhead for NDN.

Most of the proposed solutions[20,55,56,58,63,65] are based on applying an additional delay to privacy-sensitive contents. Mostly this delay is applied by the edge routers. Delay-based approaches try to maintain a record of previously received content. This record is used for selecting the content for which delay should be applied. The problems with the delay-based approach are that they degrade the NDN performance by degrading average delay, and they also use some memory. However, the delay-based approach is more effective against the cache privacy attack. Therefore, it is necessary to improve their performance by reducing average delay and memory usage.

5 Cache Pollution Attack

In this attack, the attacker sends interest packets to modify the configuration of content stored in CS of

Table 5. Cache Privacy Attack Types and Countermeasures

Reference	Attack Type	Countermeasure				
		Type	Applied by	Effectiveness	Implementation	Topology
Arianfar *et al.*[19], 2011	NWLA & CAA	Publishing encrypted chunks of file	Producer	High		
Lauinger *et al.*[54], 2012	TBA, ODA & DFCA					
Lauinger *et al.*[20], 2012	TBA & CAA	Delay-based, caching-based, & disabling packet field	Edge routers	Intermediate		
Ntuli and Han[66], 2012	ODA	Detection-based approach	Edge routers	Intermediate		
Acs *et al.*[55], 2013	TBA	Delay first k requests	Edge routers	Intermediate	CCNx codebase	Self-made
Chaabane *et al.*[56], 2013	TBA	Delay-based & caching-based	Edge routers	Intermediate		
Mohaisen *et al.*[65], 2013	TBA	Delay for time t	Edge routers	High	CCNx codebase	Linear
Gao *et al.*[57], 2015	TBA	Detection-based approach	Edge routers	Intermediate		
Mohaisen *et al.*[58], 2015	TBA	Delay for time t	Edge routers	High	CCNx codebase	Linear
Dogruluk *et al.*[60], 2016	TBA	Detection-based approach	Edge routers	Intermediate		
Abani and Gerla[62], 2016	TBA	Caching-based on betweenness centrality	Routers near core routers	Intermediate	ndnSIM	Transit-stub
Acs *et al.*[63], 2017	TBA	Delay first k requests	Edge routers	Intermediate	CCNx codebase	Self-made
Kamath *et al.*[64], 2017	TBA	Collaborative caching	Edge routers	Intermediate	ndnSIM	Self-made
Kumar *et al.*[67], 2018	TBA	Detection-based approach	Edge routers	Intermediate	ndnSIM	Linear

nearby routers. Depending upon the modification, the cache pollution attack has been classified into two types by Deng *et al.*[69]: locality disruption attack (LDA) and false locality attack (FLA). In LDA, the attacker requests interest packets for new unpopular contents. These unpopular contents get cached in CS of nearby NDN routers, thereby reducing their cache hit probability for requests corresponding to popular contents. In FLA, the attacker increases the popularity of content(s) by issuing a large number of requests for them. The goal of the attack is to create the false popularity of contents in nearby routers. This attack confuses detection approaches based only on local popularity. Fig.10 shows a demonstration of FLA. The attacker compiles a list having names of unpopular contents and then requests them one by one. These contents replace popular contents which are already presented in CS, thereby reducing the hit ratio of CS.

LDA and FLA are performed similarly, but the main difference is the intention of the attacker. In LDA, the objective of the attacker is to decrease the hit ratio of CS. Therefore, the attacker requests new unpopular contents. In FLA, the objective of the attacker is to confuse the router by creating false popularity for an unpopular content or a group of unpopular contents. The attacker repeatedly requests a set of contents for performing this attack. Both these attacks are difficult to detect as the attacker does not follow any specific pattern. Breslau *et al.*[70] showed that the web traffic follows Zipf[71] distribution. Therefore, caching of frequently requested contents yields better performance. The attacker utilizes this fact and selects contents from the tail of Zipf to perform the attack. There is a low probability that consumers will request these selected contents.

5.1 Detection and Mitigation of Cache Pollution Attack

Detection approaches for the cache pollution attack have been divided into three major categories, i.e., statistical approach, probabilistic approach, and miscellaneous approach. Approaches given in [22, 72–77] are threshold-based statistical approaches. Park *et al.*[21] proposed an entropy-based approach which comes under the probabilistic approach. Karami and Guerrero-Zapata's Adaptive Network-Based Fuzzy Inference System (ANFIS) based approach[78] is classified under the miscellaneous approach. Fig.11 shows the types of detection approaches used for cache pollution attacks.

In the following subsections work done by different authors[19,20,54–67] is discussed according to the type of detection approach used by them.

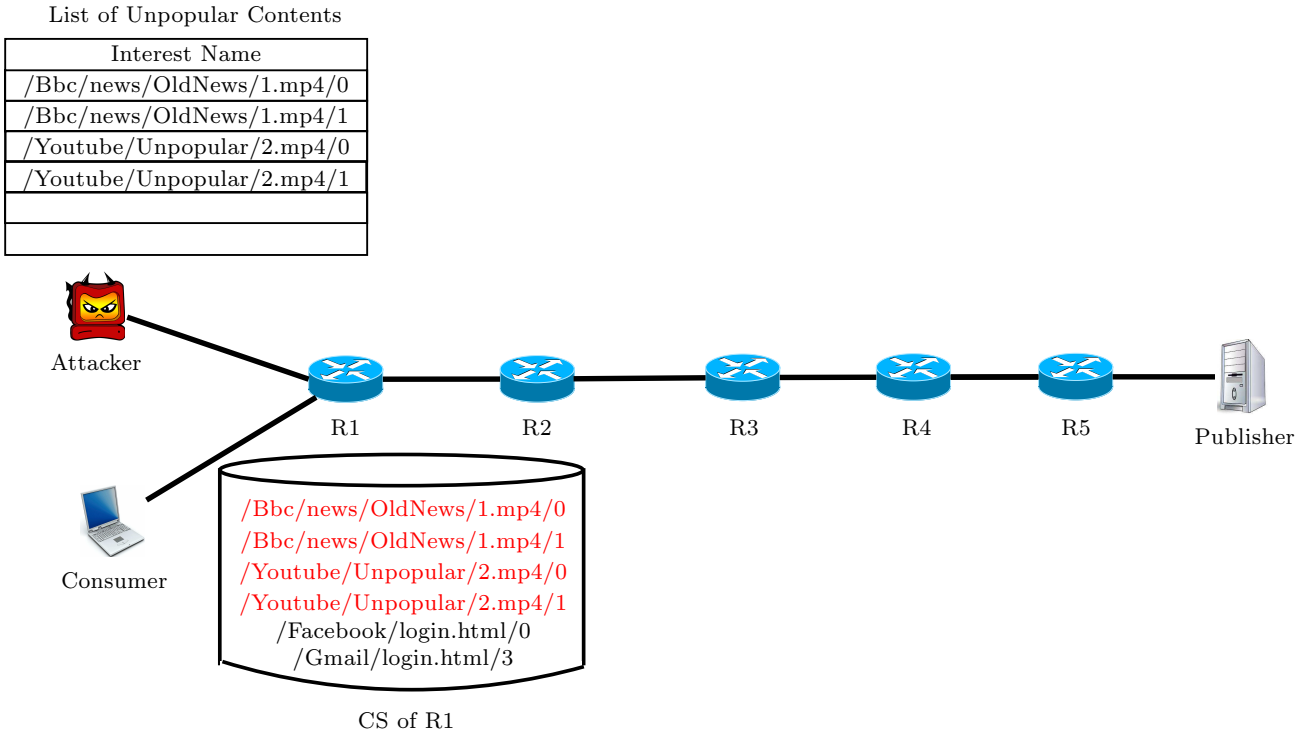

Fig.10. False locality attack.

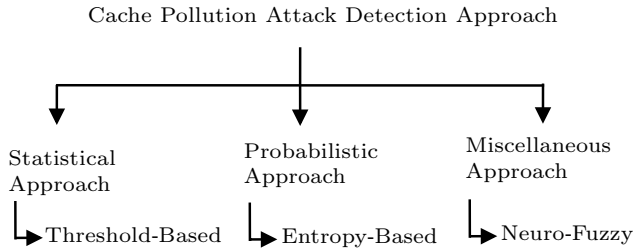

Fig.11. Types of detection for cache pollution attacks.

5.1.1 Threshold-Based Approach

Xie *et al.*[22] proposed an approach for the mitigation of cache pollution attacks called CacheShield. In this approach, the CS stores a placeholder for the name and frequency of the content along with actual content. When the router receives the content for the first time, then its placeholder is stored in CS with its count value t which is initially set to zero. If the router receives a content whose placeholder is already stored in CS, then it computes a shield function and decides whether the content will be cached or not. If the router decides to cache the content, then the router replaces the content placeholder with the actual content. Otherwise, the value t of the content placeholder is incremented. This approach has high space and time complexity as the router uses extra space for storing the name placeholder and it uses extra CPU cycle for computing shield function whenever it receives a content. Also, the authors did not discuss how they implemented caching of content and content placeholder in the same CS.

Conti *et al.*[72] proposed an approach for the detection of both LDA and LFA. This approach consists of two phases, i.e., the learning phase and the detection phase. In the learning phase, the caching algorithm creates a set S by choosing IDs of contents by random sampling. The router keeps track of the contents belonging to S in order to calculate the threshold τ, which is used for detecting the attack. In the detection phase, the router computes the variability of measurement, i.e., δ_m which depends on the frequency of content which belongs to S, the size of measurement, and the probability of occurrence of content. If δ_m is greater than τ then the attack is detected. This approach uses low memory and low computation as compared with Xie *et al.*'s approach[22] as it uses sample S whose size is small. While the approach[22] applies to LDA only, this approach can detect both types of cache pollution attack. This approach depends on the configuration of S; therefore, choosing the appropriate S is necessary.

Xu *et al.*[73] proposed an approach based on the assumption that in cache pollution attack the attacker requests a large number of interest packets for a small group of unpopular namespaces. This approach has two phases, i.e., traffic monitoring phase and identification phase. In the traffic monitoring phase, the router uses Lightweight Flajolet Martin (LFM) sketch to monitor distinct interest packets having a common namespace. Monte Carlo hypothesis test[79] is used for computing the detection threshold. In the identification phase, the router periodically monitors traffic and calculates the result. If the result exceeds the threshold, then the router announces that the attack is going on. This approach has comparatively low memory and low computational overhead. Xu *et al.*[73] mentioned that in cache pollution attack the attacker requests a large number of interest packets for the same namespace. However, legal requests can also have this type of patterns. Suppose some consumers request chunks of a newly published popular movie from the same publisher. In this case, each chunk has a unique name having the same namespace. Therefore this approach will detect this event as an attack.

Kamimoto *et al.*[74] proposed an approach to mitigate cache pollution attack based on the hierarchy of contents. This approach has three steps as follows: 1) identifying the attackers' prefixes, 2) cache recovery, and 3) cache protection. In the first step, the router calculates the weighted request rate variation per prefix (WRVP) which is used for making a black-list. In the second step, the router removes unpopular cached contents using the black-list. In the third step, the router does not cache future unpopular contents. This approach uses less memory as compared with the other approaches[21,22] as per namespace statistics is stored rather than each content. This approach has a limitation that the attacker can use unpopular contents of the popular prefix for performing the attack. These interest packets belong to a popular prefix and hence they are not listed in the black-list.

Guo *et al.*[75] proposed an approach based on the path diversity of interest packets received by a backbone router. In normal condition, contents are requested from many different regions, thereby following different router-level paths. They utilized this fact and proposed a scheme in which backbone routers trace paths of interest packets for each content object stored in its CS. The backbone router maintains two pieces of information for every content object stored in its CS, i.e., the times the content object gets hit and data structure named as PathTracker which stores the number of

distinct paths $o.path_e$ followed by interest packets corresponding to the content object. Bernoulli distribution is used to compute the expected number of distinct paths, i.e., $o.path_t$ using the value of $o.hit$. Further, the values of $o.path_e$, $o.path_t$ and $o.hit$ are used for calculating ratios r_t and r_e given by (5) and (6) respectively.

$$r_t = \frac{o.hit}{o.path_t}, \tag{5}$$

$$r_e = \frac{o.hit}{o.path_e}. \tag{6}$$

If r_e is sufficiently greater than r_t for a given object then the object is polluted. On detecting that content o is polluted, it is removed from CS.

Salah *et al.*[76] utilized the CoMon framework (discussed in Section 3) for the mitigation of cache pollution attacks. It classifies nodes into three categories, i.e., ISP controller (IC), monitoring nodes (MNs) and NDN nodes (NNs). The MNs periodically send the IC a list of contents they observed with the times the contents are requested. The IC then forms a white-list of prefixes based on their popularity. It then distributes the contents of the white-list to the MNs and instructs them to cache only those contents. NNs are informed to cache those contents which are not in the white-list of the MR to which the NNs are associated. This approach ensures that the popular data always remains cached in the network, thus reducing the number of costly Inter-ISP data requests. The authors of [76] did not give any specific technique for the distribution of content among MNs. The distribution of contents is significant as it contributes to the NDN performance. If a content is cached away from its locality, then requesting it will result in a higher delay.

Zhang *et al.*[77] proposed an approach for the detection and mitigation of cache pollution attack based on the locality of content. The decision of whether to cache or not is taken using two parameters, i.e., the coefficient of variation (COV) and popularity per prefix. When the router receives an interest packet, then the router updates its dataset, having information such as its prefix, incoming interface, and frequency. This dataset is used for calculating COV. The popularity of a prefix is the times interest packets are requested for it. When the router receives a data packet, then the router uses the value of popularity and COV for deciding whether to cache the data packet or not.

5.1.2 Entropy-Based Approach

Park *et al.*[21] proposed an approach for detecting low rate LDA leveraging entropy of requests. In this approach, a router maps the name of each content object to an entry of a binary matrix. The indices of the matrix represent the value obtained after hashing the name and taking the modulus of the result using two different positive integers. If the object is present for a particular mapping, then the corresponding value is set to 1; else it is set to 0. The rank value of matrix after applying Gaussian elimination is used for detecting the change in the random behavior of the request. EWMA is used for smoothing the value of rank over time. When the value of rank goes beyond a predefined threshold, then the attack is detected. This approach can be applied to both TCP/IP and ICN type networks as it does not use any user ID. The size of storage depends on the size of the cache which is equal to $O(\sqrt{n})$. Here n is the size of the cache. This approach uses high computation as the router has to compute mapping for each request in the table.

5.1.3 Miscellaneous Approaches

ANFIS stands for Adaptive Neuro-Fuzzy Inference System[78]. ANFIS is an advanced cache replacement policy, which uses a neuro-fuzzy network to calculate the goodness value of the data packets. ANFIS works on each router independently by collecting statistics about each data packet and then passing this information through five layers of a fuzzy network to refine the goodness value. This value is used by the cache replacement policy for making the caching decision. Though ANFIS can counter most of the attacks efficiently, it will fail to counter smartly designed attacks explicitly tailored to defeat ANFIS because, in ANFIS, each router is working completely independent of other routers. Therefore, the nearby attacker can manipulate the router's decision of caching. Since the router only has access to the local data, it may wrongly perceive the attacker's request to be legitimate and thus start caching contents requested by the attacker replacing the actual legitimate content from the cache.

5.2 Findings

Table 6 summarizes types of cache pollution attacks, their detection, and the countermeasures. Most of the work is done for the mitigation of LDA[21,22,72,73,76–78]. The mitigation of LDA is comparatively easier than the mitigation of FLA. FLA manipulates local popularity; therefore, local popularity based detection or caching schemes do not work against FLA. Most of the detection approaches[22,72–77] for cache privacy attack are based on the computed statistical-threshold. The

Table 6. Cache Pollution Attack Types and Countermeasures

Reference	Attack Type	Countermeasure				
		Detection Type	Memory Overhead	Computational Overhead	Implimentation	Topology
Park *et al.*[21], 2012	LDA	Entropy-based	Low	High		
Xie *et al.*[22], 2012	LDA	Threshold-based	High	High	CCNx codebase	Self-made
Conti *et al.*[72], 2013	Both	Threshold-based	Low	Low	ndnSIM	XC and DFN
Karami and Guerrero-Zapata[78], 2015	Both	ANFIS	Low	Low	ndnSIM	XC and DFN
Xu *et al.*[73], 2015	Both	Threshold-based	Low	Low	ndnSIM	Tree
Kamimoto *et al.*[74], 2016	FLA	Threshold-based	Low	Low	ndnSIM	XC and DFN
Guo *et al.*[75], 2016	FLA	Threshold-based	Medium	Low	Simulation	K-ary Tree
Salah *et al.*[76], 2017	Both	Threshold-based	High	Low	ndnSIM	AS-3967
Zhang *et al.*[77], 2017	LDA	Threshold-based	Low	Low	ndnSIM	AS-1221, 1755, 3967, and 7018

Note: Blank means not applicable (N.A.) as the authors did not give a detection or mitigation approach in their paper.

threshold is computed by observing the content packet received by the router and then calculating the threshold based on the observation. The countermeasure is applied by comparing the current statistics of the content with the threshold. If the content is unpopular, then it is not stored in the cache. The "memory overhead" and "computational overhead" attributes in Table 6 show the comparative amount of memory space and CPU cycles of the router used in deploying the countermeasure respectively.

6 Content Poisoning Attack

Basically, NDN has three types of contents, viz. valid contents, fake contents, and corrupted contents. The valid content has a valid signature generated by a valid publisher's private key. The fake content has a valid signature that may be generated with any private key, but it is not associated with the published namespace. The corrupted content has an invalid signature.

The content poisoning attack poisons the CS of the routers by injecting fake or corrupted contents into them. Any request for the fake or corrupted content(s) cached them in all the intermediary routers, thus leading to the spreading of poisoned content(s) across the network. The verification of such contents can be done by any consumer using the content signature, but the verification by a router working at line speed is a time-consuming process. The trust model used by different applications makes it more difficult for routers to verify contents.

Content poisoning attacks can be carried out in two ways[17]. Some of the routers can be compromised by the attacker(s) and these compromised routers spread poisoned contents in reply to the interest packets. The genuine intermediary routers cache the poisoned contents, which are further accessed by other consumers, thus poisoning the network. Fig.12 shows the demonstration of content poisoning attacks through a malicious router. Another way of attacks is based on the distribution of poisoned contents through compromised publishers and routers. The malicious content is created for replying to interest packets that are expected in a large number (for example, the request for the score of a live football match). The interest packets for legitimate contents are replied with the poisoned content by the compromised routers or producers. This attack is more critical than the attack through compromised routers only due to its capability to spread fake or corrupted contents faster. Most of the mitigation approaches for the content poisoning attack have used some fields of interest or data packets which are not presented in current NDN packet specification, i.e., exclude & Publisher Public Key Digest (PPKD), and KeyLocator. The exclude field is a field in the interest packet which is used by the consumer to exclude the content which it does not want as a reply. The PPKD field is present in both the interest and the data packets. It contains the hash of the producer's public key. The KeyLocator field is a filed of the data packet which contains either public key itself or any reference to locate the public key.

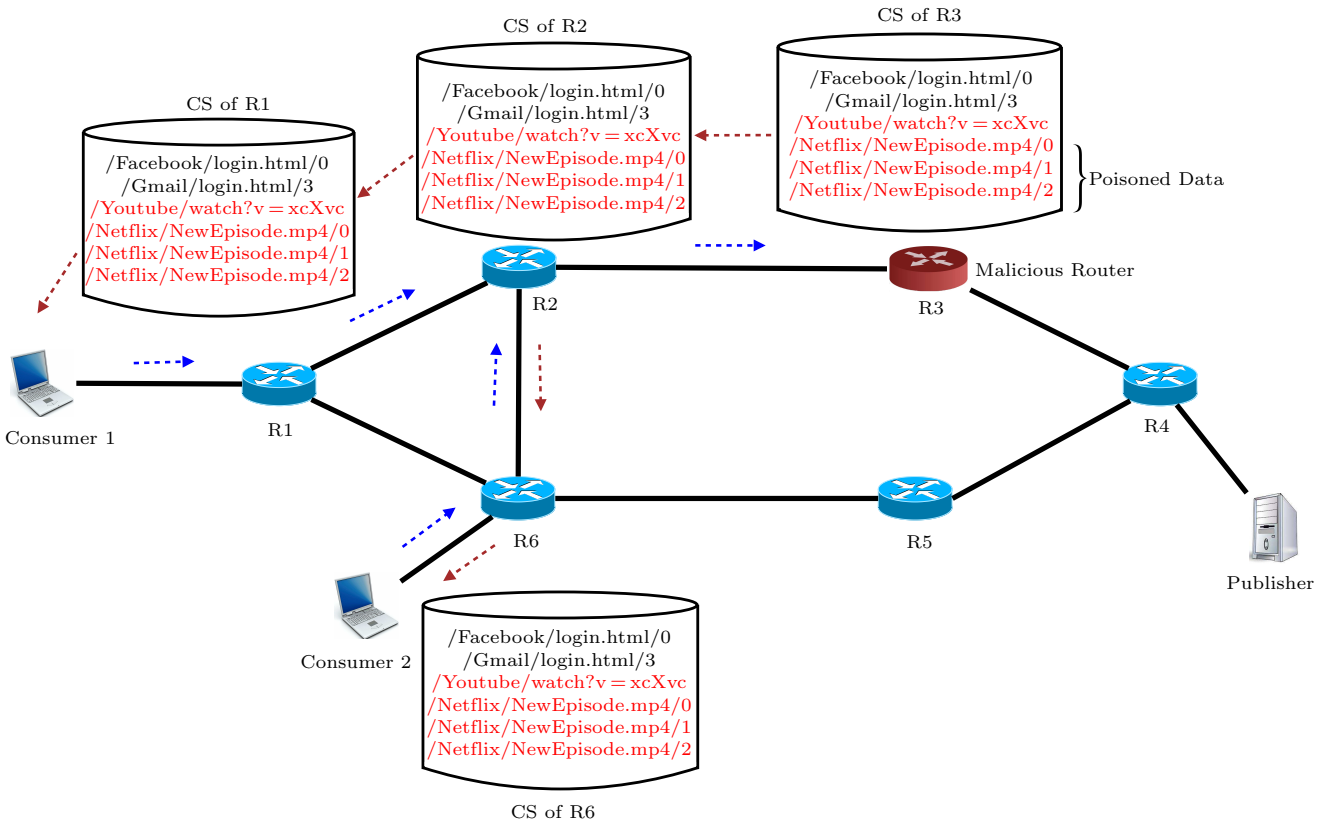

Fig.12. Content poisoning attack through a compromised router.

6.1 Detection and Mitigation of Content Poisoning Attack

The detection of content poisoning attacks can be done by checking the PPKD field or the content signature or both of them. Most of the existing approaches for the detection of attacks check these two parameters only. Apart from this, Mai *et al.*[80,81] proposed a machine learning based approach to detect the poisoned contents. Fig.13 shows the categorization of the detection approaches for the content poisoning attack.

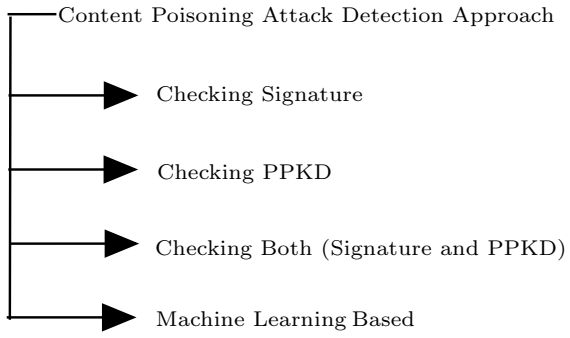

Fig.13. Content poisoning attack detection based on checking.

The detection of the poisoned content by a router is an expensive process. Therefore, to verify the content, a decision can be taken based on the information contained in the packet received by a router. Alternately, the feedback from the consumer or other routers can also be taken for deciding whether the content is poisoned or not. If the content is found poisoned, then it is evicted. The various detection and mitigation approaches for content poisoning attacks are discussed next in chronological order.

Gasti *et al.*[17] proposed a scheme called Self Certifying Interests/Content (SCIC) for verifying the content. This scheme is based on Self Certifying Name (SCN)[82], which has been utilized for the NDN architecture. SCIC introduced two naming schemes, Static SCIC (SSCIC) and Dynamic SCIC (DSCIC). In SSCIC, the interest packet contains the hash of the data packet along with the name. The router can easily verify the content by comparing the hash value of the received data packet with the hash value presented in the interest packet. However, the consumer should know the hash value of the data packet before requesting it. The authors suggested a way to link the data packets to one another, i.e., each received data packet contains

the hash value of the next data packet. This scheme creates a dependency between contents; therefore, it is not suitable for dynamic content. In DSCIC, the consumer sends interest packets containing the hash of public key in the PPKD field. The router can check the authenticity of content by matching the PPKD field of the received data packet with the PPKD field of the interest packet. This approach detects fake contents, but it cannot detect corrupted contents. For detecting corrupted contents, the authors proposed three countermeasures based on the collaborative scheme (involving both a router and a consumer), i.e., Probabilistic Disjoint Verification (PDV), Neighbour Verification Feedback (NVF), and Consumer Feedback (CF).

In PDV, content verification is distributed among n routers $(r_1, r_2, ..., r_n)$ which belong to the same autonomous system (AS). Here each r_i verifies content object CO. If $h_{CO} = i \bmod n$, here h_{CO} is the last 32 bits of the hash of CO. This scheme can be further secured by using a keyed hash function where each AS has its secret key. In NVF, a router performs content verification probabilistically and independently. When a poisoned content is found then the router sends a warning interest like "/warning/CO" to its neighbor. On receiving the warning message the neighbor performs the verification of the poisoned content given in the warning message with some probability and further informs its neighbors. In CF, each cached content has its trust value T which lies between 0 and 1. T equal to 1 means the content is verified and if T is closer to 0, then it should be selected for verification with probability proportion of $1 - T$, or deleted when the cache becomes full. Initially, a content is assigned $T = 0.5$. This value decreases on receiving the negative feedback from the consumer in the exclude field of the interest packet and increases when it is verified by the consumer's feedback using special request messages. In PDV and NVF, the router has to perform the additional computation of verifying interest which is not suitable for router processing at line speed. CF can be used by the consumer to propagate a new attack by giving the wrong knowledge to routers.

Ghali *et al.*[23] proposed a content ranking algorithm in which the content is ranked according to the exclude field of the interest packet sent by the consumer. Each content has a ranking value between 0 and 1. The new content has a ranking 1. This rank is degraded gracefully as the content is excluded by a consumer using the exclusion field. This approach is similar to the consumer feedback approach mentioned in [17], and therefore, suffers from the same limitation, i.e., a consumer may give wrong knowledge to a router using the exclude field.

Ghali *et al.*[24] mentioned some of the vulnerabilities of the NDN, i.e., neither the digest component of the name nor the PPKD field is a required field of the interest packet. NDN does not have any unified trust model which can be used by the consumer application to fetch the hash of a given content securely. The network layer trust management and the content poisoning attack are inter-dependent. Based on abovementioned vulnerabilities of the NDN Ghali *et al.*[24] proposed Interest-Key-Binding (IKB) rule for enforcing trust in NDN. According to IKB, the interest must reflect the public key of the producer. The implication of IKB for producers is that they should include the public key in the KeyLocator field of the data packet. The implication of IKB for the router is that it should hash the public key of the received content and match it with the PPKD field of a corresponding PIT entry. If a mismatch occurs, the content is discarded. Otherwise, the content signature is verified. And if it is valid then the content is forwarded and cached. Else, the content is dropped. IKB consequence for a consumer is the need to obtain and validate the producer's public key before issuing the interest for any content originated by the producer. Authors[24] showed that if the IKB rule is followed then content poisoning is impossible.

Nam *et al.*[83] performed ns-3-based simulation on a self-made topology and showed that 90% of traffic is composed of bypassing contents. Based on the above observation, they gave a scheme based on verifying signature only in the case of a cache hit. To apply this scheme in NDN, they used Segmented Least Recently Used (SLRU) as a cache replacement policy. In SLRU, the cache is divided into two segments, i.e., protected and unprotected. LRU applies to both segments. If a content object is hit, then its signature is verified. If the content is not a poisoned content, then it is moved from the unprotected segment to the protected segment. The contents of the protected segment are preferred over the contents of the unprotected segment. Thus the verified content has more chances to be accessed again and again. Kim *et al.*[84] extended the work in [83] to present its mathematical analysis. The authors further extended their work in [85] to propose a new type of attack on their previous work[84] called verification attack. In this attack, the adversary initially feeds the CS of the compromised routers with a large number of polluted contents. Then the adversary requests these con-

tents using the compromised hosts. The authors suggested that the compromised router would serve more content as compared with a normal router. Based on the above assumption Kim *et al.*[85] proposed an approach for this attack which consists of two phases, i.e., the detection phase and the identification phase. In the detection phase, the ratio of the number of contents hit to the number of verifications of the contents is used as a parameter for the detection of the attack. A dynamic threshold (τ) has been chosen as per (7).

$$\tau = \min(1, \zeta \frac{W(H_p)}{W(H)}). \tag{7}$$

Here ζ is an EWMA function, $W(H_p)$ is the probability that the content that has been inserted into the CS by a cache-miss generates a cache-hit for the next corresponding request, and $W(H)$ is the cache-hit probability of all the contents. After the detection phase, the identification phase is triggered in which the router detects malicious faces by monitoring the number of verified contents forwarded per face.

DiBenedetto *et al.*[86] proposed an approach based on the feedback of the consumer application. When a poisoned content is detected, the network stack automatically generates an interest packet having the information of the poisoned content. After receiving this interest packet the router acts according to one of the two forwarding strategies as follows: Immediate Failover and Probe First. In the first approach, the detecting node makes the malicious face least preferred for the future. In the second approach, the node stops forwarding the interest packets for the namespace(s), which are under attack. Additionally, the node alerts all next hop routers of this malicious namespace.

Wu *et al.*[87] proposed an approach for the mitigation of content poisoning attack based on the feedback reputation score (FRS) of the router. This FRS depends on the ratio of copies of poisoned content to the total number of copies of a given content also called poisoning ratio (PR). FRS can be defined by (8)[87].

$$FRS_{kj} = \begin{cases} 1, & \text{if } C_i \text{ is not poisoned,} \\ -\theta \times e^{\alpha(1-PR(C_i))}, & \text{if } C_i \text{ is poisoned.} \end{cases} \tag{8}$$

Here C_i is the content forwarded from router j to router k. θ specifies the amount of punishment imposed on router j. α measures the intensity of punishment. Each router j calculates reputation value (RV) of its neighbor k using (9).

$$RV_{kj}I(t) = \delta \times RV_{kj}(t-1) + \sum_{h=1}^{N} FRS_{kj}. \tag{9}$$

Here N is the number of contents which comes from router j to router k in time interval $[t-1, t]$ and $t \geqslant 1$. Initially $RV_{kj}(0) = 0$. A router receiving the interest packet for content C_i forwards it to the next hop router having a higher RV. If a content C_i is forwarded from a router or producer to another router, then the router stores hash (C_i) and face ID on which C_i is received in the PIT for some time. When the consumer receives C_i, then he/she first authenticates C_i. If C_i is poisoned then the consumer sends verification result (vr_i) and hash (C_i) to the upstream router. The router receiving this verification message compares the received hash with the already stored hash and increments the counter corresponding to the hash of the content. This counter is used for calculating PR which can be further used for computing RV via (9).

Nguyen *et al.*[88] found three vulnerabilities in NDN, i.e., unregistered remote provider, multicast forwarding, and best route forwarding. The cause of the first vulnerability is that the data packet received by any of the faces can act as a reply to the interest packet. Therefore, the content may be satisfied by a malicious producer before it could be satisfied by a genuine producer. In multicast forwarding, the interest packet is forwarded to all the faces registered to corresponding FIB entry. Therefore, there is a possibility that a malicious producer can reply to the interest packet. In the best route forwarding, a router ignores a similar interest packet having the same name and selectors but different nonce when it is received during retransmission suppression interval. The interest received after this interval will be forwarded through the next lowest-cost face; thus a malicious producer has a chance to satisfy the interest packet with a poisoned content.

Mai *et al.*[80] proposed a detection approach for content poisoning attack based on Bayesian Network (BN). The data for the detection of content poisoning attack is obtained by Montimage Monitoring Tool (MMT) which monitors 18 different matrices like CS hit, CS miss, and PIT exist time. Traffic corresponding to attack and non-attack scenario is collected which is used to train the BN. The trained BN is used for the detection of the attack. Authors showed that the accuracy of the detector increased from 53% up to 93% by raising the detection window from 1 s to 5 s. In [81], the authors

implemented the detection module using Docker containers in the OpenStack platform. The NDN nodes having NDN firewalls and NDN signature verification modules are hosted on virtual machines (VMs), which are managed by Docker containers. BN is implemented using a python library for Bayesian network models called pgmpy. If the attack is detected, then the reaction is triggered in which the NDN firewall is informed to block this malicious content. This approach uses a trained BN which is trained using the data obtained by a fixed topology and traffic scenario. Therefore the BN has to be trained in every new situation (topology, traffic, etc.).

Hu *et al.*[89] proposed an approach for the mitigation of content poisoning attack which utilizes name-key based forwarding and multipath forwarding. This approach has three phases, i.e., route building phase, normal content retrieval phase, and recovery phase. In the route building phase, the producer advertises the route advertisement along with the PPKD and its authorization. This advertisement is accepted by the routing system which authenticates it and installs FIB entries in routers. In the normal content retrieval phase, the consumer requests the content by sending the interest packet having the name corresponding PPKD. On receiving the content, the consumer verifies its signature. If it is a poisoned content, then the consumer resends the interest packet excluding the name of the poisoned content. The router receiving the interest packet forwards it using multipath forwarding.

Further, the router verifies the content given in the exclude field of the interest packet and evicts it if it is a poisoned content. In this approach, the consumer has to know the PPKD of content at the time of requesting it. Thus this approach needs a system that distributes PPKDs of the contents to the consumers. It uses extra memory for storing extra PPKD field in FIB, PIT, and CS.

6.2 Findings

Table 7 summarizes the content poisoning attack. In this attack, the attacker can be a router, a publisher or a consumer, unlike the attackers in the interest flooding attack, the cache privacy attack, or the cache pollution attack. The attack is difficult to detect due to the maliciousness of the routers. The attacker can bluff the detection-based approach for content poisoning attacks as it does not influence network resources. The incapability of the router to verify the signature of each content at line speed for different trust models further complicates the attacks. Therefore, in most of the approaches[17,23,86,87,89], the verification of the content

Table 7. Proposed Solutions to Content Poisoning Attack Detection

Reference	Checked by	Detection by Checking	Proposed Solution	Simulator	Topology
Gasti *et al.*[17], 2013	Consumer & router	PPKD & Signature	SSCIC & DSCIC		
Ghali *et al.*[23], 2014	Consumer	Signature	Content Ranking Algorithm	ndnSIM	Tree-based, AT & T, and DFN
Ghali *et al.*[24], 2014	Router	First PPKD then signature	IKB		
Nam *et al.*[83], 2015	Router	Signature in case of cache-hit	Extention of SLRU	ndnSIM	Self-made
Kim *et al.*[84], 2015	Router	Signature in case of cache-hit	Extention of SLRU	ndnSIM	Self-made
DiBenedetto and Papadopoulos[86], 2016	First consumer then router	Signature	Modifying forwarding strategy	ndnSIM	SprintPoP
Wu *et al.*[87], 2016	First consumer then router	Signature	Forwarding based on RV	ndnSIM	European backbone
Kim *et al.*[85], 2017	Router	Signature in case of cache-hit	Extention of SLRU	ndnSIM	Self-made
Mai *et al.*[80], 2018	Router	18 different matrices	BN-based classifier	NFD	Linear
Mai *et al.*[81], 2018	Router	18 different matrices	BN-based classifier	NFD over Docker container	Self-made
Hu *et al.*[89], 2018	Consumer & router	PPKD & signature	Name-key based forwarding & multipath forwarding	ndnSIM	Self-made

signature is done by the consumer. Some articles[24,89] recommend that consumers should specify PPKD of the content with the interest packet. But, this approach needs a third party or search engine like Google. The router can store the specified PPKD in the PIT, and the replied content can be checked by matching the PPKD presented in the content and PIT entry. Though this approach can check fake contents, it cannot check corrupted contents. After checking PPKD, the content signature can reduce the number of signature checking by the router[89]. Nam *et al.*[83−85] reduced the number of signatures checked by the router by checking the signature only in case of cache hit. The machine learning based approach by Mai *et al.* in [80] cannot adapt successfully as per new attack.

7 Open Research Challenges in NDN Security

The open research issues in NDN make the detection and mitigation of the NDN attacks more challenging. These issues pertain to the challenges in the deployment of mitigation algorithms and the effects of deploying mitigation algorithms in the NDN architecture. The research issues corresponding to each of the four attacks presented in this article have been categorically listed in the following subsections.

7.1 Research Issues Corresponding to Interest Flooding Attack

1) The detection approach for the interest flooding attack must have parameters that can differentiate between attack and non-attack scenarios.

2) The parameters used for the detection should be fine-grained so as to detect the malicious interface as well as the malicious namespace. Hence, the countermeasure could be applied to the specific pair of interface and namespace.

3) The countermeasure should have an initial push-back mechanism, followed by a targeted trace-back. The initial push-back mechanism helps to lower the effect of the interest flooding attack and the trace-back helps to detect and block the attacker.

4) The detection and mitigation of new types of interest flooding attack, such as bIFA, cIFA, and collusive interest flooding attack should be done.

5) A single approach should be developed to restrict all types of interest flooding attack.

6) The detection and mitigation approach should be capable of adopting against any new type of interest flooding attack.

7.2 Research Issues Corresponding to Cache Privacy Attack

1) The contents should be decisively checked about whether they are private or not, which is a complicated problem.

2) Another crucial issue which should be handled is the determination of authority (viz. producer, consumer, or a third party) that decides the privacy of the contents.

3) The effective detection of cache privacy attack must be done as it does not affect network resources and is not easily detectable.

4) Most of the mitigation approaches for cache privacy attacks are based on applying an additional delay before giving the content from the CS. The mitigation approach with low average delay should be designed.

5) The computation and storage requirements for mitigation approaches should be low as routers have limited resources.

7.3 Research Issues Corresponding to Cache Pollution Attack

1) The effective determination of the content's popularity should be done by choosing appropriate popularity matrices.

2) Popularity matrices should consider the local and global popularity, as local popularity may be affected by the attackers.

3) The calculation of a content's global popularity is complicated but should be done for better mitigation approaches.

4) The mitigation approach with low computation, low storage, and low message overhead should be designed.

5) The detection-based approach should have high accuracy and low detection time.

6) The determination and penalization of the attackers should be done.

7.4 Research Issues Corresponding to Content Poisoning Attack

1) The mechanism for routers to quickly check the signature of the content at line speed should be designed.

2) The attacker can be a router, a producer, or a compromised consumer. Mitigation approaches should consider all the possibilities.

3) The consumers can easily verify the signature of the content. The consumer feedback should be utilized at the router to detect poisoned contents.

4) Mitigation approaches should reduce the number of signatures that need to be checked by the router.

8 Conclusions

NDN is one of the most promising candidates among all the ICNs. It is more secure than the current TCP/IP model because of its content-based security. However, new types of attacks are possible in NDN. This paper reviews four major kinds of security attacks in NDN, interest flooding attack, cache privacy attack, cache pollution attack, and content poisoning attack. The various aspects of the interest flooding attack such as interest flooding attack types, detection techniques, the granularity and parameters for the detection, and mitigation approaches are addressed. The different types of cache privacy attack, their countermeasures, the entity which applies the countermeasure, and the effectiveness of the countermeasures are explained. The types of cache pollution attack, their detection techniques, and the memory and computation used by the countermeasure are discussed. For the content poisoning attack, the detection type is presented based on the checking of the signature or PPKD. The proposed solutions and the entity which applies the solution are also elaborated. Finally, the research issues corresponding to each of the four attacks have been highlighted. This article is an attempt to provide researchers the domain knowledge of the specific issues in NDN security. The presented open research issues motivate the researchers to work in this area.

References

[1] Wein J M, Kloninger J J, Nottingham M C *et al*. Content delivery network (CDN) content server request handling mechanism with metadata framework support. US Patent, 2017. http://www.freepatentsonline.com/20180109489.pdf, June 2019.

[2] Barkai D. Peer-to-Peer Computing: Technologies for Sharing and Collaborating on the Net (1st edition). Intel Press, 2002.

[3] Özsu M T, Valduriez P. Principles of Distributed Database Systems (3rd edition). Springer Science & Business Media, 2011.

[4] Ahlgren B, Dannewitz C, Imbrenda C, Kutscher D, Ohlman B. A survey of information-centric networking. *IEEE Communications Magazine*, 2012, 50(7): 26-36.

[5] Koponen T, Chawla M, Chun B G *et al*. A data-oriented (and beyond) network architecture. In *Proc. the ACM SIGCOMM 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, August 2007, pp.181-192.

[6] García-de-Blas G, Beben A, Ramón F J, Maeso A, Psaras I, Pavlou G *et al*. COMET: Content mediator architecture for content-aware networks. In *Proc. the 2011 Future Network & Mobile Summit*, June 2011, Article No. 25.

[7] Marc M, Solis I, Wood C A. Content-centric networking-architectural overview and protocol description. https://arxiv.org/abs/1706.07165, Oct. 2019.

[8] Zhang L, Afanasyev A, Burke J *et al*. Named data networking. *ACM SIGCOMM Computer Communication Review*, 2014, 44(3): 66-73.

[9] Zhang L, Estrin D, Burke J *et al*. Named Data Networking (NDN) project. Technical Report, Xerox Palo Alto Research Center-PARC, 2010. https://www.cs.arizona.edu/∼bzhang/paper/ndn-tr.pdf, June 2019.

[10] Hoque A, Amin S O, Alyyan A, Zhang B, Zhang L, Wang L. NLSR: Named-data link state routing protocol. In *Proc. the 3rd ACM SIGCOMM Workshop on Information-Centric Networking*, August 2013, pp.15-20.

[11] Afanasyev A, Zhu Z, Yu Y, Wang L, Zhang L. The story of ChronoShare, or how NDN brought distributed secure file sharing back. In *Proc. the 12th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, Oct. 2015, pp.525-530.

[12] Zhu Z, Bian C, Afanasyev A, Jacobson V, Zhang L. Chronos: Serverless multi-user chat over NDN. Technical Report NDN-0008, Named Data Networking Project Team, 2012. http://www.named-data.net/techreport/TR008-chronos.pdf, June 2019.

[13] Zhang H. NDNFit: An open mHealth application built on Named Data Networking [Ph.D. Thesis]. University of California, 2018.

[14] Gusev P, Burke J. NDN-RTC: Real-time videoconferencing over named data networking. In *Proc. the 2nd ACM Conference on Information-Centric Networking*, September 2015, pp.117-126.

[15] Afanasyev A, Shi J, Zhang B *et al*. NFD developer's guide. Technical Report, 2014. https://users.cs.fiu.edu/∼afanasyev/assets/papers/tr-afanasyev2018nfd-dev-guide.pdf, June 2019.

[16] Zhang Z, Yu Y, Zhang H *et al*. An overview of security support in Named Data Networking. *IEEE Communications Magazine*, 2018, 56(11): 62-68.

[17] Gasti P, Tsudik G, Uzun E, Zhang L. DoS and DDoS in Named Data Networking. In *Proc. the 22nd International Conference on Computer Communication and Networks*, July 2013, Article No. 67.

[18] Afanasyev A, Mahadevan P, Moiseenko I, Uzun E, Zhang L. Interest flooding attack and countermeasures in Named Data Networking. In *Proc. the 2013 IFIP Networking Conference*, May 2013, Article No. 7.

[19] Arianfar S, Koponen T, Raghavan B, Shenker S. On preserving privacy in content-oriented networks. In *Proc. the 2011 ACM SIGCOMM Workshop on Information-Centric Networking*, August 2011, pp.19-24.

[20] 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 Computer Communication Review*, 2012, 42(5): 54-57.

[21] Park H, Widjaja I, Lee H. Detection of cache pollution attacks using randomness checks. In *Proc. the 2012 IEEE International Conference on Communications*, June 2012, pp.1096-1100.

[22] Xie M, Widjaja I, Wang H. Enhancing cache robustness for content-centric networking. In *Proc. the 2012 IEEE INFOCOM*, March 2012, pp.2426-2434.

[23] Ghali C, Tsudik G, Uzun E. Needle in a haystack: Mitigating content poisoning in Named-Data Networking. In *Proc. the 2014 NDSS Workshop on Security of Emerging Networking Technologies*, February 2014, Article No. 5.

[24] Ghali C, Tsudik G, Uzun E. Network-layer trust in named-data networking. *ACM SIGCOMM Computer Communication Review*, 2014, 44(5): 12-19.

[25] Saxena D, Raychoudhury V, Suri N, Becker C, Cao J. Named Data Networking: A survey. *Computer Science Review*, 2016, 19: 15-55.

[26] Chen S, Mizero F. A survey on security in Named Data Networking. arXiv:1512.04127, 2015. https://arxiv.org/abs/1512.04127, June 2019.

[27] Dai H, Wang Y, Fan J, Liu B. Mitigate DDoS attacks in NDN by interest traceback. In *Proc. the 2013 IEEE Conference on Computer Communications Workshops*, April 2013, pp.381-386.

[28] Signorello S, Marchal S, François J *et al.* Advanced interest flooding attacks in Named-Data Networking. In *Proc. the 16th IEEE International Symposium on Network Computing and Applications*, October 2017, pp.1-10.

[29] Salah H, Strufe T. Evaluating and mitigating a collusive version of the interest flooding attack in NDN. In *Proc. the 2016 IEEE Symposium on Computers and Communication*, June 2016, pp.938-945.

[30] Yi C, Afanasyev A, Moiseenko I, Wang L, Zhang B, Zhang L. A case for stateful forwarding plane. *Computer Communications*, 2013, 36(7): 779-791.

[31] Compagno A, Conti M, Gasti P, Tsudik G. Poseidon: Mitigating interest flooding DDoS attacks in Named Data Networking. In *Proc. the 38th Annual IEEE Conference on Local Computer Networks*, October 2013, pp.630-638.

[32] Tang J, Zhang Z, Liu Y, Zhang H. Identifying interest flooding in Named Data Networking. In *Proc. the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, August 2013, pp.306-310.

[33] Wang K, Zhou H, Qin Y, Chen J, Zhang H. Decoupling malicious interests from pending interest table to mitigate interest flooding attacks. In *Proc. the 2013 Global Communications Conference*, December 2013, pp.963-968.

[34] Vassilakis V G, Alohali B A, Moscholios I, Logothetis M D. Mitigating distributed denial-of-service attacks in Named Data Networking. In *Proc. the 11th Advanced International Conference on Telecommunications*, June 2015, pp.18-23.

[35] Wang K, Zhou H, Qin Y, Zhang H. Cooperative-filter: Countering Interest flooding attacks in Named Data Networking. *Soft Computing*, 2014, 18(9): 1803-1813.

[36] Nguyen T N, Cogranne R, Doyen G, Retraint F. Detection of interest flooding attacks in Named Data Networking using hypothesis testing. In *Proc. the 2015 IEEE International Workshop on Information Forensics and Security*, November 2015. Article No. 18.

[37] Xin Y, Li Y, Wang W, Li W, Chen X. A novel interest flooding attacks detection and countermeasure scheme in NDN. In *Proc. the 2016 IEEE Global Communications Conference*, December 2016, Article No. 43.

[38] Zhi T, Luo H, Liu Y. A Gini impurity-based interest flooding attack defence mechanism in NDN. *IEEE Communications Letters*, 2018, 22(3): 538-541.

[39] Ding K, Liu Y, Cho H H, Chao H C, Shih T K. Cooperative detection and protection for interest flooding attacks in Named Data Networking. *International Journal of Communication Systems*, 2016, 29(13): 1968-1980.

[40] Xin Y, Li Y, Wang W, Li W, Chen X. Detection of collusive interest flooding attacks in Named Data Networking using wavelet analysis. In *Proc. the 2017 IEEE Military Communications Conference*, October 2017, pp.557-562.

[41] Karami A, Guerrero-Zapata M. A hybrid multiobjective RBF-PSO method for mitigating DoS attacks in Named Data Networking. *Neurocomputing*, 2015, 151: 1262-1282.

[42] Kumar N, Singh A K, Srivastava S. Evaluating machine learning algorithms for detection of interest flooding attack in Named Data Networking. In *Proc. the 10th International Conference on Security of Information and Networks*, October 2017, pp.299-302.

[43] Kumar N, Singh A K, Srivastava S. Feature selection for interest flooding attack in Named Data Networking. *International Journal of Computers and Applications*. doi:10.1080/1206212X.2019.1583820.

[44] Li Z, Bi J. Interest cash: An application-based countermeasure against interest flooding for dynamic content in Named Data Networking. In *Proc. the 9th International Conference on Future Internet Technologies*, June 2014, Article No. 2.

[45] Alston A, Refaei T. Neutralizing interest flooding attacks in Named Data Networks using cryptographic route tokens. In *Proc. the 15th IEEE International Symposium on Network Computing and Applications*, October 2016, pp.85-88.

[46] Salah H, Wulfheide J, Strufe T. Coordination supports security: A new defence mechanism against interest flooding in NDN. In *Proc. the 40th IEEE Conference on Local Computer Networks*, October 2015, pp.73-81.

[47] Salah H, Wulfheide J, Strufe T. Lightweight coordinated defence against interest flooding attacks in NDN. In *Proc. the 2015 IEEE Conference on Computer Communications Workshops*, April 2015, pp.103-104.

[48] Mallat S. A Wavelet Tour of Signal Processing (2nd edition). Academic Press, 1999.

[49] Wang L, Pan Y, Dong M, Yu Y, Wang K. Economic levers for mitigating interest flooding attack in Named Data Networking. *Mathematical Problems in Engineering*, 2017, 2017: Article No. 4541975.

[50] Rokach L, Maimon O. Top-down induction of decision trees classifiers — A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (*Applications and Reviews*), 2005, 35(4): 476-487.

[51] Zhao L, Cheng G, Hu X *et al.* An insightful experimental study of a sophisticated interest flooding attack in NDN. In *Proc. the 1st IEEE International Conference on Hot Information-Centric Networking*, August 2018, pp.121-127.

[52] Afanasyev A, Moiseenko I, Zhang L. ndnSIM: NDN simulator for NS-3. Technical Report, University of California, 2012. https://named-data.net/wp-content/uploads/TRndnsim.pdf, June 2019.

[53] Lauinger T. Security & scalability of content-centric networking [Master Thesis]. Technische Universität Darmstadt, Darmstadt, 2010.

[54] 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, 2012. http://mail.seclab.tuwien.ac.at/papers/ccn-cache-attacks-iseclab-0812-001.pdf, June 2019.

[55] Ács G, Conti M, Gasti P, Ghali C, Tsudik G. Cache privacy in Named-Data Networking. In *Proc. the 33rd IEEE International Conference on Distributed Computing Systems*, July 2013, pp.41-51.

[56] Chaabane A, de Cristofaro E, Kâafar M A, Uzun E. Privacy in content-oriented networking: Threats and countermeasures. *ACM SIGCOMM Computer Communication Review*, 2013, 43(3): 25-33.

[57] Gao M, Zhu X, Su Y. Protecting router cache privacy in Named Data Networking. In *Proc. the 2015 IEEE/CIC International Conference on Communications in China*, November 2015, Article No. 23.

[58] Mohaisen A, Mekky H, Zhang X, Xie H, Kim Y. Timing attacks on access privacy in information centric networks and countermeasures. *IEEE Transactions on Dependable and Secure Computing*, 2015, 12(6): 675-687.

[59] Compagno A, Conti M, Gasti P, Mancini L V, Tsudik G. Violating consumer anonymity: Geo-locating nodes in Named Data Networking. In *Proc. the 13th International Conference on Applied Cryptography and Network Security*, June 2015, pp.243-262.

[60] Dogruluk E, Costa A, Macedo J. Evaluating privacy attacks in Named Data Network. In *Proc. the 2016 IEEE Symposium on Computers and Communication*, June 2016, pp.1251-1256.

[61] Lutz R. Security and privacy in future Internet architectures-benefits and challenges of content centric networks. arXiv:160101278, 2016. https://arxiv.org/abs-/1601.01278, June 2019.

[62] Abani N, Gerla M. Centrality-based caching for privacy in information-centric networks. In *Proc. the 2016 IEEE Military Communications Conference*, November 2016, pp.1249-1254.

[63] Ács G, Conti M, Gasti P, Ghali C, Tsudik G, Wood C. Privacy-aware caching in information-centric networking. *IEEE Transactions on Dependable and Secure Computing*, 2017, 16(2): 313-328.

[64] Kamath A A, Jamadagni C, Anilkumar A, Mathew K, Tahiliani M P. GCPiN: Group caching for privacy in Named Data Networking. In *Proc. the 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems*, December 2017, Article No. 68.

[65] Mohaisen A, Zhang X, Schuchard M, Xie H, Kim Y. Protecting access privacy of cached contents in information centric networks. In *Proc. the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, May 2013, pp.173-178.

[66] Ntuli N, Han S. Detecting router cache snooping in Named Data Networking. In *Proc. the 2012 International Conference on Information and Communication Technology Convergence*, October 2012, pp.714-718.

[67] Kumar N, Singh A K, Srivastava S. A triggered delay-based approach against cache privacy attack in NDN. *International Journal of Networked and Distributed Computing*, 2018, 6(3): 174-184.

[68] DiBenedetto S, Gasti P, Tsudik G, Uzun E. ANDaNA: Anonymous Named Data Networking application. arXiv:11-122205, 2011. https://arxiv.org/abs/1112.2205, June 2019.

[69] Deng L, Gao Y, Chen Y, Kuzmanovic A. Pollution attacks and defenses for internet caching systems. *Computer Networks*, 2008, 52(5): 935-956.

[70] Breslau L, Cao P, Fan L, Phillips G, Shenker S. Web caching and Zipf-like distributions: Evidence and implications. In *Proc. the 18th Annual Joint Conference of the IEEE Computer and Communications Societies*, March 1999, pp.126-134.

[71] Zipf G K. Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology (Kindle Edition). Ravenio Books, 2016.

[72] Conti M, Gasti P, Teoli M. A lightweight mechanism for detection of cache pollution attacks in Named Data Networking. *Computer Networks*, 2013, 57(16): 3178-3191.

[73] Xu Z, Chen B, Wang N, Zhang Y, Li Z. ELDA: Towards efficient and lightweight detection of cache pollution attacks in NDN. In *Proc. the 40th IEEE Conference on Local Computer Networks*, October 2015, pp.82-90.

[74] Kamimoto T, Mori K, Umeda S, Ohata Y, Shigeno H. Cache protection method based on prefix hierarchy for content-oriented network. In *Proc. the 13th IEEE Annual Consumer Communications Networking Conference*, January 2016, pp.417-422.

[75] Guo H, Wang X, Chang K, Tian Y. Exploiting path diversity for thwarting pollution attacks in Named Data Networking. *IEEE Transactions on Information Forensics and Security*, 2016, 11(9): 2077-2090.

[76] Salah H, Alfatafta M, SayedAhmed S, Strufe T. CoMon++: Preventing cache pollution in NDN efficiently and effectively. In *Proc. the 42nd IEEE Conference on Local Computer Networks*, October 2017, pp.43-51.

[77] Zhang G, Liu J, Chang X, Chen Z. Combining popularity and locality to enhance in-network caching performance and mitigate pollution attacks in content-centric networking. *IEEE Access*, 2017, 27(5): 19012-19022.

[78] Karami A, Guerrero-Zapata M. An ANFIS-based cache replacement method for mitigating cache pollution attacks in Named Data Networking. *Computer Networks*, 2015, 80: 51-65.

[79] Gilks W R, Richardson S, Spiegelhalter D. Markov Chain Monte Carlo in Practice (1st edition). Chapman and Hall/CRC, 1996.

[80] Mai H L, Nguyen T, Doyen G *et al*. Towards a security monitoring plane for Named Data Networking and its application against content poisoning attack. In *Proc. the 2018 IEEE/IFIP Network Operations and Management Symposium*, April 2018, Article No. 133.

[81] Mai H L, Aouadj M, Doyen G *et al*. Implementation of content poisoning attack detection and reaction in virtualized NDN networks. In *Proc. the 21st Conference on Innovation in Clouds, Internet and Networks and Workshops*, Feb. 2018, Article No. 14.

[82] Mazières D, Kaminsky M, Kaashoek M F, Witchel E. Separating key management from file system security. In *Proc. the 17th ACM Symposium on Operating Systems Principles*, December 1999, pp.124-139.

[83] Nam S, Kim D, Yeom I. Content verification in Named Data Networking. In *Proc. the 2015 International Conference on Information Networking*, January 2015, pp.414-415.

[84] Kim D, Nam S, Bi J, Yeom I. Efficient content verification in Named Data Networking. In *Proc. the 2nd International Conference on Information-Centric Networking*, September 2015, pp.109-116.

[85] Kim D, Bi J, Vasilakos A V, Yeom I. Security of cached content in NDN. *IEEE Transactions on Information Forensics and Security*, 2017, 12(12): 2933-2944.

[86] DiBenedetto S, Papadopoulos C. Mitigating poisoned content with forwarding strategy. In *Proc. the 2016 IEEE Conference on Computer Communications Workshops*, April 2016, pp.164-169.

[87] Wu D, Xu Z, Chen B, Zhang Y. What if routers are malicious? Mitigating content poisoning attack in NDN. In *Proc. the 2016 IEEE Trustcom/BigDataSE/ISPA*, August 2016, pp.481-488.

[88] Nguyen T, Marchal X, Doyen G, Cholez T, Cogranne R. Content poisoning in Named Data Networking: Comprehensive characterization of real deployment. In *Proc. the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management*, May 2017, pp.72-80.

[89] Hu X, Gong J, Cheng G, Zhang G, Fan C. Mitigating content poisoning with name-key based forwarding and multipath forwarding based inband probe for energy management in smart cities. *IEEE Access*, 2018, 6: 39692-39704.

Naveen Kumar has done his Bachelor of Technology (computer science and engineering) from U.P. Technical University, Lucknow, in 2012. He has done his Master of Technology (software engineering) in the Department of Computer Science and Engineering at MNNIT (Motilal Nehru National Institute of Technology) Allahabad, Prayagraj, India, in 2014. He is currently pursuing his Ph.D. in the Department of Computer Science and Engineering at MNNIT Allahabad, Prayagraj. His research interests include peer-to-peer systems, future internet technologies, network security, and Named Data Networking.

Ashutosh Kumar Singh has done his Bachelor of Technology (information technology) from U.P. Technical University, Lucknow, in 2011. He has done his Master of Technology (computer science and engineering) from Indian Institute of Information Technology and Management, Gwalior, India, in 2014. Currently, he is a Ph.D. student in the Department of Computer Science and Engineering, MNNIT (Motilal Nehru National Institute of Technology) Allahabad, Prayagraj. He is a member of ACM and IEEE. His research interest includes network optimization and Software-Defined Networking.

Abdul Aleem has done his Bachelor of Technology (computer science and engineering) from U.P. Technical University, Lucknow. He taught for three years in academic institutes after his graduation. He has done his Master of Technology (software engineering) in the Department of Computer Science and Engineering at MNNIT (Motilal Nehru National Institute of Technology) Allahabad, Prayagraj. He was bestowed Gold Medal for being the topper of his batch in the Masters program. He had served as lead engineer in Samsung Research Institute (SRI), Noida, India, for more than four years. Currently, he is pursuing his Ph.D. degree in the Department of Computer Science and Engineering at MNNIT Allahabad, Prayagraj. His research interest is in the area of data mining, machine learning, and educational systems.

Shashank Srivastava has done his Bachelor of Technology (computer science and engineering) from U.P. Technical University, Lucknow. He got his M.S. degree in information security from Indian Institute of Information Technology Allahabad, Prayagraj. He obtained his Ph.D. degree in information technology from Indian Institute of Information Technology Allahabad, Prayagraj, in 2014. He is currently working as an assistant professor in the Department of Computer Science and Engineering, MNNIT (Motilal Nehru National Institute of Technology) Allahabad, Prayagraj. He is a member of ACM, IEEE, CSI and CRSI (Cryptographic Research Society of India). His areas of expertise are Software Defined Networking (SDN), Named Data Networking (NDN), network flow optimization and security, information security, and future Internet technologies.