Security Performance and DDoS Attacks within NDN Environment

Nawel Kortas
Prince Research Laboratory, ISITCom,
University of Sousse, Tunisia

ABSTRACT

Named-Data Networking (NDN) is one such effort that exemplifies the content-centric approach to networking. Rather than naming locations (i.e., hosts or interfaces), NDN names content, which becomes first-class entity. This permits decoupling of content from the host that strength store and/or disseminate it, facilitating automatic caching and optimizing bandwidth custom. Due to its new architecture, NDN introduces new security and privacy challenges. These challenges include data privacy, anonymity, access control and authentication. It also includes some basic security features. However, NDN's flexibility to DDoS attacks has not been analyzed to date. In this paper, we present a specific and concrete scenario of DDoS attack in NDN. We also set mechanisms that defend against DDoS attacks such as signature and network defense.

Keywords

NDN, DDos, CCN, CDN, IP, Security, Integrity, Future Internet Architectures.

1. INTRODUCTION

Named Data Networking (NDN) [1] [3] is a research project that is developing the future Internet architecture using the principles behind of CCN. Security and privacy are among the fundamental requirements for NDN. In current Internet, Distributed Denial of Service (DDoS) attacks consume the resources of a remote host or network, thereby denying or degrading service to legitimate users. Therefore, NDN's resilience to DDoS attacks deserves our full attention.

This paper studies a specific and concrete scenario of NDN attacks. Our goal is to present and set practical and general resolution of security mechanisms.

This paper is organized as follows. Section 2 puts the main characteristics of NDN. Section 3 presents the related works. Section 4 presents the impact of current attacks on NDN, and set the initial analysis of NDN's resilience to DDoS attacks. Finally, Section 5 concludes the paper.

2. NDN

2.1 Overview

NDN [4] is an on-going research project that aims to evolve it into an architectural framework for the future Internet. It is also considered an instance of the broader Information Centric (ICN) approach to networking. NDN explicitly names content (data) instead of physical locations and thus transforms content into a first-class entity. It also stipulates that each piece of named content must be digitally signed by its producer. This allows for decoupling of trust in content from trust in the entity that might store hat content.

2.2 Why NDN?

There are many methods to send packets, many protocols to ensure the format of the content, and many applications that creates and receives packets. However, there is only IP.

IP does not provide all services. Since there is no restriction about data encryption, data in packet may be easily read by packets forwarders if the packet creators did not do anything about security server. Traffic near server is incredibly busy. That is because each request has a destination address to a particular; NDN was designed to solve all those problems.

NDN realigns the architecture with application needs by adopting named data as the thin waist of the hourglass architecture. NDN greatly simplifies application development, and new applications in turn will drive further growth and success of the future Internet. It also offers a very strong notion of secure end-to-end data transmission. NDN networks are able to self-regulate traffic flows for both unicast and multicast traffic without relying on transport protocols. Moreover, NDN separates routing schemes and forwarding mechanisms. It facilitates choice and competition by tolerating users as shown by a network economic model [6].

2.3 How it works?

In NDN, clients tell the network what they need instead of using the network to send requests to servers. Clients do not need to know server's IP address. Clients ask nearby routers for certain data packets by sending interest packet. In IP network, applications provide their content. For NDN, applications name its data. Interest packets are routed based on the name of data in the packets. Once a router receives a packet, if there is no useful data in its cache, the interest packet will be added to routers pending interest table before the router forward it, else if the interest packet reaches appropriate producer, the producer will send data packet to the router who provide the interest packet

2.4 Unities and differences between IP and NDN

Both architectures share the same hourglass shape, with the IP/NDN layer as the narrow waist. Also, they send datagrams and follow end-to-end principle. They use their own namespace for data delivery (IP uses IP addresses to deliver datagrams between IP nodes; NDN uses the application name space to deliver datagrams between NDN nodes).

In today's global routing system, IP uses only a single path to each destination, and that path is often asymmetric due to "hot-potato" routing. It is difficult to measure and compare performance. They use a different name space: IP address vs name. NDN includes a security primitive directly at the narrow waist (every Data packet is signed). IP sends packets to destination addresses; NDN uses Interest packets to fetch Data packets. IP has a stateless data plane. NDN has a stateful data plane. Together with the forwarding strategy, this stateful data plane offers NDN networks a variety of desired functions.

2.5 NDN architecture

An NDN Data packet is meaningful independent of where it comes from or where it may be forwarded to, thus the router can cache it to satisfy potential future requests. This enables NDN to automatically support various functionality without extra infrastructure, including content distribution (many users requesting the same data at different times), multicast (many users requesting the same data at the same time), mobility (users requesting data from different locations), and delay-tolerant networking (users having intermittent connectivity).

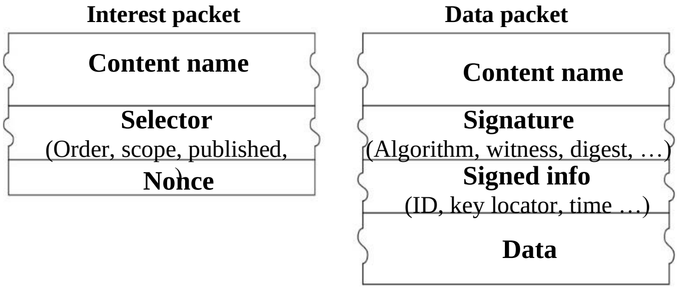

Figure 1: Packets in the NDN Architecture.

Below we describe some elements of the NDN architecture.

- **Names**

NDN names are opaque to the network; routers do not know the meaning of a name (although they know the boundaries between components in a name). This allows each application to choose the naming scheme that fits its needs and allows the naming schemes to evolve independently from the network.

- **Data-Centric Security**

In NDN, security is built into data itself, rather than being a function of where, or how, it is obtained [5]. Each piece of data is signed together with its name, securely binding them. Data signatures are mandatory applications cannot “opt out” of security.

The signature, coupled with data publisher information, enables determination of data provenance, allowing the consumer’s trust in data to be decoupled from how (and from where) data is obtained. It also supports fine-grained trust, allowing consumers to reason about whether a public key owner is an acceptable publisher for a particular piece of data in a specific context.

- **Routing and Forwarding**

NDN routes and forwards packets on names, which eliminates four problems that addresses pose in the IP architecture: address space exhaustion, NAT traversal, mobility, and scalable address management. There is no address exhaustion problem since the namespace is unbounded. There is no NAT traversal difficult since a host does not need to expose its address in order to offer content. Mobility, which requires changing addresses in IP, no longer breaks communication since data names remain the same. Finally, address assignment and management is no longer required in local networks, which is especially empowering for sensor networks.

- **Caching**

Upon receiving an Interest, an NDN router first checks the Content Store. If there is a data whose name falls under the Interest’s name, the data will be sent back as a response. The Content Store, in its basic form, is just the buffer memory in today’s router. Both IP routers and NDN routers buffer data packets. The difference is that IP routers cannot reuse the data after forwarding them, while NDN routers are able to reuse the data since they are identified by persistent names. For static files, NDN achieves almost optimal data delivery. Even dynamic content can benefit from caching in the case of multicast or packet retransmission after a packet loss. Cache management and replacement is subject to ISP (Internet Service Provider).

- **Pending Interest Table (PIT)**

The PIT contains the arrival interfaces of Interests that have been forwarded but are still waiting for matching Data. This information is required to deliver data to their consumers. To maximize the usage of the PIT, PIT entries need to be timed out pretty quickly, somewhere around packet round-trip time. However, if they are timed out prematurely, Data will be dropped, and it is the consumer’s responsibility to retransmit his/her Interests.

- **Transport**

The NDN architecture does not have a separate transport layer. It moves the functions of today’s transport protocols up into applications, their supporting libraries, and the strategy component in the forwarding plane. Multiplexing and demultiplexing among application processes is done directly using names at the NDN layer, and data integrity and reliability are directly handled by application processes where the appropriate reliability checking, data signing and trust decisions can be made.

2.6 Comparison of ICN, CCN, CDN and NDN

The term “Information-Centric Networking” (ICN) [3] appeared around 2010, likely inspired by Van Jacobson’s 2006 Google Tech Talk “A New Way to look at Networking“. This talk points out a new direction of moving the Internet toward content distribution architecture. ICN represents a broad research direction of content/information/data centric approach to network architecture. NDN is a specific architecture design under the broad ICN umbrella.

CCN refers to the architecture project Van started at PARC, which included leading the development of software codebase that represents a baseline implementation of this architecture. The NDN project originally used CCNx as its codebase, but as of 2013 has forked a version to support the needs specifically related to the NSF-funded architecture research and development.

A CDN [2] is a good example of service that is implemented as an overlay on today’s TCP/IP architecture to meet the demand for scalable content distribution, when the same content is requested by many users. CDNs operate at the application layer, which gives rise to two issues: how to get customer content requests into the CDN system and mapping each request to the nearest CDN node serving the content. NDN works directly at network layer and naturally forwards Interest packets along the best paths to the desired data.

2.7 DNS no longer needed in NDN networks

NDN networks no longer need the “DNS name to IP address” look up service. However as a globally deployed distributed database, today’s DNS has been used for a variety of other purposes besides mapping domain names to IP addresses. We are currently exploring the potential to use a distributed database system similar to DNS to address routing scalability and other issues.

3. RELATED WORK

As shown in [7], it’s difficult to detect timing attacks exploit NDN routers as “oracles” and allow the adversary to learn whether a nearby consumer recently requested certain content.

First, it is suggested that consumers and producers should indicate which content is privacy-sensitive. Second, several techniques are provided to balance certain tradeoffs between privacy and latency. A formal model is also introduced that allows quantifying the degree of privacy offered by various caching algorithms. It is shown in [8] that cache pollution attack is a realistic threat on NDN. The researchers have conducted experiments to confirm that the attacks previously demonstrated on very small topologies can extend on larger and more realistic networks with no additional effort. They point out that existing proactive countermeasures are ineffective against realistic adversaries. Also detecting and limiting the attack may prove to be a better strategy. Their simulations show that the lightweight detection technique provides accurate results. In [9], the authors aim to raise awareness of privacy attacks as an intrinsic and relevant issue in NDN architectures. They argue that the tradeoff between privacy and performance can be balanced at several layers of abstraction:

- Whether certain protocol features should be allowed,
- At what aggregation level caches should be placed,
- What content may be cached?

Given an approach to classify objects according to their sensitivity, the most fine-grained one is to leave the major non-sensitive traffic unaffected and to prevent privacy sensitive content from being cached. In [10], the authors propose an approach that detects cache snooping attempts targeted low-level routers. Their detecting algorithm takes input: the network graph (g), the candidate selection function (f), the trust function (t). The trust function is a mapping of trustworthiness between two connected nodes in the graph. A node satisfying f is defined as a candidate. A candidate that further satisfies t is called a snooper. When the algorithm initiates, it creates two empty sets: one set contains snoopers and the other one contains candidates. For each node in graph G, f is used to select candidates. The trust function t is called to determine snoopers from candidates. In function t, a candidate with computed trustworthiness less than the threshold k is regarded as a snooper. The output of the above procedure is the set of detected snoopers out of candidates. Such algorithm combines formal signatures with trust systems and pattern recognition to increase the level of confidence in snooper detection.

4. What is a DDoS attack?

Attacks against distributed networks are also known as Distributed Denial of Service (DDoS) attacks. This type of attack takes advantage of specific capacity limits that apply to network resources, such as the infrastructure that supports a company's website. A DDoS attack involves sending multiple requests to the web resource under attack in an attempt to interfere with the website's ability to handle requests and block its operation. Main targets of DDoS attacks:

- Online shopping sites
- Online Casinos
- Companies or organizations providing online services

4.1 How does a DDoS attack work?

Network resources, such as web servers, can only handle a limited number of requests at the same time. Besides the capacity limit of the server, the channel that connects the server to the Internet also has a limited bandwidth/capacity. When the number of requests exceeds the maximum capacity of an infrastructure component, the service level may experience the following issues:

- Response to queries is much slower than normal.
- Requests from some or all users can be completely ignored.

Typically, the attacker's goal is to block the functioning of the web resource (full denial of service). The attacker can also demand money to stop the attack. In some cases, a DDoS attack may even take the form of an attempt to discredit or harm a competitor's business.

To send an extremely large number of requests to the targeted resource, the cybercriminal often establishes a “botnet” of infected computers. Since the attacker controls the actions of each of the infected computers in the botnet, the scale of the attack can overwhelm the victim's web resources.

Sometimes botnets, with their networks of compromised devices, are hired out for other potential attacks through “attack for hire” services. This allows people with bad intentions but without training or experience to easily launch DDoS attacks on their own.

4.2 Types of DDoS Attacks

There are many different types of DDoS attacks, and hackers often use several to inflict havoc on their targets. The three main types of attacks are volumetric, protocol, and application layer attacks. The goal of all attacks is to dramatically slow down or prevent legitimate traffic from reaching its destination. For example, it may be preventing a user from accessing a website, purchasing a product or service, watching a video, or interacting on social media. Additionally, by rendering resources unavailable or degrading performance, DDoS attacks can cause business to stop. This can result in employees not being able to access email, web applications, or conduct business as usual.

To better understand how DDoS attacks work, let's look at the different routes hackers can take. The Open Systems Interconnection (OSI) model is a layered structure for various networking standards. It contains seven different layers. Each layer of the OSI model serves a unique purpose, like the floors of an office building that house the various functions of a business. Attackers target different layers depending on the type of web or internet resource they want to disrupt.

Protocol attacks attempt to consume and exhaust the computational capacity of various network infrastructure resources, such as servers or firewalls, through malicious connection requests that exploit protocol communications. SYN (synchronization) floods and Smurf-style DDoS attacks are two common types of protocol-based DDoS attacks. Protocol attacks can be measured in packets per second (pps) as well as bits per second (bps).

4.3 How to Interrupt a DDoS Attack?

During mitigation, the DDoS protection provider deploys a series of countermeasures aimed at stopping and reducing the impact of a distributed denial of service attack. With today's attacks becoming more sophisticated, cloud-based DDoS attack mitigation protection helps deliver defense-in-depth at scale, keeping backbone infrastructure and internet services available and operating optimally. With DDoS attack protection services, businesses can:

- Reduce the attack surface and business risks associated with DDoS attacks
- Avoid business-impacting service disruptions
- Avoid taking web pages offline
- Accelerate response to a DDoS event and optimize incident response resources

- Reduce the time needed to understand and investigate a service interruption
- Avoid any decrease in employee productivity
- Quickly deploy defensive countermeasures against a DDoS attack
- Avoid damaging brand reputation and bottom line
- Maintain application uptime and performance across the entire digital estate
- Limit costs associated with web security
- Defend against extortion, ransomware and other new and evolving threats

5. SECURITY AND PRIVACY

A fundamental security primitive is embedded in the "thin waist" of NDN: the name in each NDN packet is bound to packet content with a signature. This basic feature provides data integrity and origin authentication, as well as machinery to support trust and provenance by mapping between the packet signer and its source (e.g., an individual or an organization). Named and signed content also forms a more solid foundation for building secure applications, but poses two major scaling challenges: cost-effective fine-grained signature operations, and functional and usable trust management infrastructure.

5.1 Impact of Current Attacks on NDN

- **Reflection Attacks**

A reflection attack involves three parties [12]: the adversary, a victim host, and a set of secondary victims (reflectors). The goal of the adversary is to use the reflectors to overwhelm the victim host with traffic. To do so, a reflection attack uses IP packets with forged addresses: the adversary replaces its own source address with the address of its intended victim, and sends these packets to the secondary victims. Responses to such packets are not routed back to the adversary, and overwhelm the victim instead. To be effective, such attacks require some form of amplification, i.e., the amount of data used by the adversary to perform the attack must be significantly smaller than the amount of data received by the victim. NDN is generally resilient to this type of attack due to the symmetric nature of the path taken by each interest and the corresponding content. A content packet must follow, in reverse, the path established by the preceding interest. However, note that an NDN router is allowed to broadcast an incoming interest on some or all of its interfaces. (In other words, an interest broadcast can occur at any hop).

- **Bandwidth Depletion**

In a typical coordinated distributed attack, adversary-controlled zombies' flood their victims with IP traffic in order to saturate their network resources. The usual goal is to make the victims unreachable by others and/or, more generally, to inhibit victims' ability to communicate. Normally, such attacks are carried out via TCP, UDP or ICMP and rely on sending a stream of packets to the victim at the maximum data rate. 2 A similar kind of attack can be mounted against NDN by directing a large number of zombies to request existing content from a certain victim. However, it is easy to see that the effectiveness of this attack would be very limited. Once the content is initially pulled from its producer, it is cached at intervening routers and subsequent interests retrieve it from these routers' caches. Therefore, the network itself would limit the number of interests that reach the victim.

- **DNS Cache Poisoning**

DNS Cache Poisoning. In the current Internet, DNS servers translate human-readable names to the corresponding IP address and vice-versa. For performance reasons, DNS servers usually store the output of previous requests in their cache. There is a well know attack, called DNS cache poisoning [13], which allows the adversary to insert corrupted entries in a DNS server's cache in order to control the server responses for a set of DNS names. The best countermeasure against this attack is the use of the DNS Security Extensions protocol, i.e., DNSSEC [13]; however, as of today DNSSEC has not been widely deployed on the Internet. Packet names in NDN are routed directly, rather than being converted to addresses. While this implies that there is no need for services that perform name resolution (and therefore such service cannot be corrupted), it still is possible to conceive an attack analogous to DNS poisoning on NDN. We believe that the closest counterpart of DNS cache poisoning in NDN is a combination of route hijacking and content poisoning: the adversary would force a routing change (if necessary) that allows it to be on the path for a set of namespaces that are going to be affected by the attack. Then, it answers interests with data packets carrying an arbitrary payload.

- **Content/cache poisoning**

The adversary's goal is to cause routers to forward and cache corrupted or fake data packets, consequently preventing consumers from retrieving legitimate content. We say that a data packet is corrupted if its signature is invalid. A data packet is false if it has a valid signature, however, generated with a wrong (private) key. As mentioned in Section II, all data packets in NDN are signed. This provides the following security guarantees:

- *Integrity* : a valid signature guarantees that the signed data packet is intact;
- *Origin Authentication* : since a signature is uniquely bound to the public key of the signer, anyone can verify whether content originates with the claimed producer;
- *Correctness:* a signature binds data packet name with its payload, thus allowing a consumer to securely determine whether a data packet is a "correct answer" for the interest that requested it.

Consumers are expected to perform signature verification on every data packet before accepting it. Also, any NDN router can elect to perform signature verification for any content it forwards and caches. Upon receiving and identifying a corrupted or false data packet, a consumer can re-request a different (possibly valid) copy of the same data packet using the Exclude field in NDN interest packet.

Content signatures also trigger the issue of global trust management architecture. Without it, routers cannot determine the public key needed to verify the data packet signature. This creates a tension between flexibility (since an application can adopt an arbitrary trust model for its content) and security (any NDN router must be able to verify any data packet's signature). Even though each NDN data packet contains a reference to its signature verification (public) key, such references cannot be trusted as they can be easily abused by the adversary.

5.2 Efficiency of Signatures

NDN deals with content authenticity and integrity by making digital signatures mandatory for all content. A signature binds content with its name, and provides origin authentication no matter how, when or from where it is retrieved. Public keys are treated as regular content. NDN does not mandate any particular certification infrastructure, relegating trust management to individual applications.

Content objects are named data packets. 1 Fields of a data packet include [7]:

- *Signature*: public key signature (e.g., RSA or DSA) computed over the entire data packet, including its name.
- *Keylocator*: references the key needed to verify the content signature. This field can contain one of the following: (1) verification (public) key; (2) certificate containing verification key; or (3) NDN name referencing verification key.
- *PublisherPublicKeyDigest:* hash of the data packet producer's public key. In addition to the name of requested content, an interest packet carries several fields [8]. In this paper, we are interested in the following.
- *PublisherPublicKeyDigest*: this l field contains the hash of the producer's public key for the requested piece of data.
- *Exclude*: an optional field that embodies a description of name components that should not appear in the data packet in response to the interest.
- *AnswerOriginKind:* encodes determines whether the answer to an interest can retrieved from a CS or must be generated by the producer.
- *Scope:* limits where the Interest may propagate; Scope 0 and 1 limit propagation to the originating host; Scope 2 limits propagation to no further than the next host.

Fortunately, recent research suggests that per-packet RSA signatures for real-time data (e.g. voice) are practical on commodity end-user platforms today [11]. We could suggest another signature mechanism such as Homomorphic encryption which allowed specific types of computations to be carried out on cipher-text and generate an encrypted result. The most common use of encryption is to provide confidentiality by hiding all useful information about the plaintext. To address this we could design cryptosystems that support a variety of computations on encrypted data, ranging from general-purpose router to special-purpose router. If the RSA public key is modulus m and exponent e, then the encryption of a message x is given by $\boldsymbol{E(x) = x^e}$ mod m [14]. The homomorphic property is then:

$$\boldsymbol{E(x_1).E(x_2) = (x_{1.}\ x_2)^e \ mod\ m = E\ (x_{1.}\ x_2)}\ [14]$$

Verification cost will likely be the most important factor among signature-related challenges, since a signature is generated once but may be verified many times.

5.3 Usable Trust Management

Signature verification of NDN content merely indicates that it was signed with a particular key. Making this information useful to applications requires managing trust allowing content consumers to determine acceptable signature keys in a given context. NDN provides an excellent platform for deploying both accepted and new trust management models. Keys can be treated as named NDN data and signed NDN data items effectively function as certificates. NDN can express secure links between pieces of content [16], allowing certification of not only keys, but of content itself. This provides a rich substrate where many pieces of linked "evidence" can support consumer trust in a particular piece of content. For example, a consumer might verify the front page of the New York Times because it is signed with a well-known certified key. She can then verify individual articles because the front page links securely to them. One advantage of NDN is that it does not require a "one size fits all" trust model: trust is end-to-end, between producer and consumer. Different consumers and different content may require varying levels of assurance. However, to make NDN accessible and deployable, it must come "out of the box" with a set of usable trust mechanisms applicable to a wide range of applications.

5.4 Network Security and Defense

The research challenges for NDN [15] network security are designing a trust model to defend against attacks on the routing mesh while supporting common providers' practices and policies, and designing defenses against new types of attacks. We will design trust models appropriate to each of our routing research approaches, and implement and evaluate them in prototype routing components and experimental deployments. We will address Interest Flooding Attacks (mirroring traditional denial of service (DDoS) attacks) which send large numbers of new and distinct interests that cannot be aggregated or satisfied from caches, and Content Pollution Attacks which introduce malicious content purporting to match legitimate requests. For Interest Flooding Attacks, we plan to conduct experiments with routers that throttle the number of unsatisfied interests they will hold for a given target domain. For Content Pollution Attacks, the consumer should always use signature verification to reject malicious content, but we also plan to evaluate the burden of ingress filtering and egress filtering in (non-core) routers to protect against simulated attacks. We recognize other possible attacks, such as "hiding" content from legitimate requesters and abusing cryptographic operations to mount DDoS attacks, which we hope to enable other researchers to investigate.

6. CONCLUSION AND FUTURE WORK

In this paper, we present an overview of NDN. We tried to perform initial analysis of NDN's resilience to DDoS attacks. In doing so, we start by considering attacks on the current Internet and assess their impact on NDN. We also tried to present mechanisms that defend against DDoS attacks such as signature and network defense.

NDN needs new information theory to support reasoning about ICN networks. Much more work is needed to evaluate the effectiveness of proposed countermeasures. In particular, extensive simulation and testbed based experiments must be conducted in order to determine optimal parameters for the instantiations of our countermeasures. Finally, we intend to assess how other content-centric architectures fare with respect to DDoS attacks.

7. REFERENCES

[1] Paolo Gasti, Gene Tsudik, Ersin Uzun and Lixia Zhang,'DoS & DDoS in Named Data Networking' (2013).

[2] Wang, L.J., Lv, Y.Q., Moiseenko, I., Wang, D.S, 'A dataflow-oriented programming interface for named data networking'. J. Comput. Sci. Technol. 33, 158–168 (2018).

[3] L. Zhang et al., 'Named data networking (ndn) project', University of California and Arizona, Palo Alto Research Center and others, Tech. Rep., October (2010).

[4] Rai, S.DD., Sharma, K, 'A survey on detection and mitigation of distributed denial-of-service attack in named data networking', Advances in communication, cloud, and Big Data lecture notes in networks and systems 31 (2019).

[5] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. 'Networking named content'. In Proceedings of the 5th ACM International Conference on Emerging

Networking Experiments and Technologies, pages 1–12, (2009).

[6] Paul Laskowski and John Chuang. ‘Network monitors and contracting systems: competition and innovation’, SIGCOMM, pages 183–194, New York, NY, USA, (2006).

[7] Acs, G., M. Conti, P. Gasti, C. Ghali and G. Tsudik ‘Cache Privacy in Named-Data Networking’, ICDCS. (2013).

[8] Conti, M., Gasti, P., & Teoli, M. ‘A lightweight mechanism for detection of cache pollution attacks in Named Data Networking’, Computer Networks, 57(16), (2013).

[9] URL:http://doi.acm.org/10.1145/2378956.2378966 Comput Commun (2012).

[10] Ntuli, N. and S. Han. ‘Detecting router cache snooping in Named Data Networking’. ICT Convergence (ICTC), 2012 International Conference on, IEEE, (2012).

[11] Nguyen, T., Mai, H., Cogranne, R., Doyen, G., Mallouli, W., Nguyen, L., El Aoun, M., Montes De Oca, E., Festor, O, ‘Reliable detection of interest flooding attack in real deployment of named data networking’. IEEE Trans. Inform. Forens. Sec. 14(9), 2470–2485 (2019).

[12] Paolo Gasti, Gene Tsudik, Ersin Uzun and Lixia Zhang, ‘DoS & DDoS in Named-Data Networking’, (2012).

[13] David Dagon, Manos Antonakakis, Kevin Day, Xiapu Luo, ‘Architectures and vulnerability implications in Network and Distributed System Security Symposium’ (NDSS09), (2009).

[14] ‘A fully homomorphic encryption shema’, Craig Gentry Sep (2009).

[15] Mohammad Alhisnawi & Mahmood Ahmadi, ‘Detecting and Mitigating DDoS Attack in Named Data Networking’, Journal of Network and Systems Management volume 28, pages1343–1365 (2020).

[16] Ahmed, S.H., Bouk, S.H., Kim, D., Rawat, D.B., Song, H.: Named data networking for software defined vehicular networks. IEEE Commun. Magaz. 55(8), 60–66 (2017).