On the Scalability and Effectiveness of a Cache Pollution based DoS Attack in Information Centric Networks

Jeffery Gouge
School of Computing
University of North Florida
Jacksonville, FL 32224

Anand Seetharam
Computer Science Program
California State University Monterey Bay
Seaside, CA 93955

Swapnoneel Roy
School of Computing
University of North Florida
Jacksonville, FL 32224

***Abstract*—With the exponential growth of content, the Internet is undergoing a transformation from a host-centric approach to a content-centric one, popularly known as an *Information Centric Networks* (ICN). ICN aims to improve user performance by incorporating in-network caching at storage enabled nodes. In this paper, we explore the scalability and effectiveness of a targeted denial of service attack (DoS) designed for ICN [1]. In this attack, malicious nodes periodically request unpopular content, thereby replacing popular content in the caches enroute to the custodian with unpopular ones. The intuition behind this attack is that legitimate requests for the evicted content cannot be served from enroute caches and have to be forwarded towards the custodian, thus degrading user performance. Our goal in this paper is not to propose a sophisticated attack, but to investigate the scalability and potency of this brute force attack in ICN. By performing exhaustive and rigorous experimentation on realistic Internet topologies, and by exploring a wide range of simulation parameters we observe the following: (i) this attack is moderately successful in small scale networks comprising of less than 100 nodes, (ii) the potency of the attack rapidly decreases, and becomes ineffective as the network size increases to few thousand nodes, and (iii) the attack is more effective against a FIFO caching policy in comparison to LRU. Our results demonstrate that if the entire Internet or large MANETs are transformed into an ICN, this attack is likely to be unsuccessful.**

I. Introduction

In recent years, with the exponential increase in content, users are interested in obtaining a particular content (e.g., a video) and may not be concerned with the location (e.g., Youtube, Dailymotion) where the content is hosted. Treating content as first-class citizen, a flexible network architecture called Information Centric Networking (ICN) has been proposed which aims to evolve the current Internet from a host-centric model to a content-centric one. At its heart, ICN incorporates in-network caching at storage enabled routers to ease congestion in core networks and deliver content efficiently. In-network caching is also being adopted to improve performance of mobile ad-hoc networks (MANETs) and cellular networks.

Most recent effort in ICN has been devoted to implementing prototypes of ICN architectures [2] and designing novel routing and cache management policies to improve user level performance [3], [4]. One area of research that has received limited attention is security and privacy. Our work builds on top of existing work [1], where the authors propose a simple targeted denial of service (DoS) attack in ICN aimed at degrading user performance by polluting in-network caches with ‘unpopular’ content. By replacing popular content in intermediate caches with ‘unpopular’ content, the attack aims to degrade user performance by forcing larger fraction of requests to be satisfied by the custodian (origin server).The authors in [1] claim that the attack is successful via simulation of small networks of 9 and 30 nodes respectively and also propose countermeasures to mitigate the impact of the attack.

Our results. *The goal of this paper is not to propose a sophisticated DoS attack, but rather to investigate the scalability and effectiveness of the above mentioned attack and to explore scenarios where the attack is likely to be successful/unsuccessful.*

- We conduct extensive simulations on a number of different topologies including real Internet topologies available from the SNAP database [5]. Specifically, we test the effectiveness of the attack on grid topologies comprising of 25 and 100 nodes respectively and real peer-to-peer Gnutella topologies comprising of approximately 6500 and 9000 nodes. We evaluate the potency of the attack against different caching policies (namely LRU, FIFO and Random) and over a wide range of simulation parameters.
- Our simulations demonstrate that while the attack is moderately successful in small scale networks comprising of less than 100 nodes, its potency rapidly decreases and it becomes ineffective as the network size increases to few thousand nodes. We also observe that the FIFO caching policy is more susceptible to the attack in comparison to the LRU caching policy as it is harder to evict popular content from LRU caches in comparison to FIFO caches. Overall our simulations show that if the entire Internet or large MANETs are transformed into an ICN by enabling storage at all intermediate routers, this attack is likely to be unsuccessful.

The rest of the paper is organized as follows. We provide an overview of related work in Section II. We describe the cache pollution based DoS attack in Section III and comment on the scalability and effectiveness of the attack via extensive simulations in Section IV. We conclude the paper with discussion of future work in Section V.

II. Related Work

In this section, we provide an overview of security and privacy research in ICN. ICN being a relatively new area of research, most effort has been focused on developing efficient routing and cache management policies [3], [4]; only few prior work have explored issues related to security and privacy [6], [7], [1]. A recent survey [6] provides an exhaustive coverage of security related research in ICN. The authors in [7] enumerate DoS and DDoS attacks in ICN and suggest some possible countermeasures. In [8], the authors propose DoS attacks based on interest flooding and suggest possible countermeasures.

Our work is closest to [1], where the authors propose a cache pollution based DoS attack. The authors conclude that this attack is successful based on simulations of small networks (mainly a 9 node and 30 node network). They also propose a strategy to detect cache pollution attacks. The authors in [1] show that an existing method for detecting cache pollution called Cache Shield [9] does not work efficiently for a 30 node network. Karami in this doctorate thesis [10] investigates the use of computational intelligence for mitigating the impact of cache pollution attacks.

In contrast to prior work, we investigate the scalability and impact of a cache pollution based DoS attack similar to the one proposed in [1]. Our exhaustive simulations on multiple networks of varying sizes demonstrate that such an attack may not be effective overall. While such attacks might be effective in small networks, the impact quickly diminishes and has negligible effect on large Internet graphs consisting of thousands of nodes. One of the main reasons is that while an adversary may insert unpopular content in a cache, the content is quickly removed as the overall rate of legitimate requests for popular content is way higher than for unpopular content.

In [11], the authors explore privacy concerns in ICN; they investigate the extent to which an adversary can infer whether an user has accessed some content recently and propose solutions to mitigate the attack. In [12], the authors propose and implement a prototype for delivering content securely in ICN. Secure content naming is explored in [13], while timing attacks related to access privacy and solutions exploring the cost-benefit tradeoff are investigated in [14].

III. Problem Statement

A. Network Model

Let us consider an information centric network of N cache-enabled nodes. We assume that the content universe is of size K and that content is distributed among M custodian nodes within the network. The content custodians are equivalent to an origin server where the content is permanently hosted. When a requester requests a content, the content is forwarded towards the custodian following Dijkstra's shortest path algorithm. If a node enroute to the custodian has the content cached, it serves the request, otherwise the request is served by the custodian. While the content is being downloaded along the same path as followed by the request, it is cached on all nodes downstream from the source (enroute node or custodian) towards the requester. The intermediate routers can adopt any cache replacement policy; we experiment with LRU, FIFO and Random as these are the most widely adopted policies. Like prior work, we assume that content popularity follows a Zipfian distribution with parameter α. [1], [4]. We also assume that requests follow an independent request model (IRM).

B. Generic Attack Methodology

In this paper, we investigate the scalability of a simple DoS attack based on cache pollution, similar to the one outlined in [1]. The cache pollution based DoS attack leverages the long tail of the Zipfian distribution. A Zipfian distribution implies that a small number of content are is more popular than majority of content. We assume that some nodes in the network have been compromised (attacker nodes) and they send attack requests along with their legitimate requests. The attacker chooses K' unpopular content and requests them periodically. As mentioned in [1], as the number of unpopular content is way higher than the popular content, it does not matter if the adversary does not exactly pick the least popular content. The intuition behind this proposed attack is that the popular content in these caches is replaced with the unpopular content. As popular content is evicted from the enroute caches, the legitimate requests for these content have to routed to the custodian. As legitimate requests have to traverse additional hops, the user performance is degraded. In the following sections, we evaluate the impact of this attack via extensive simulations.

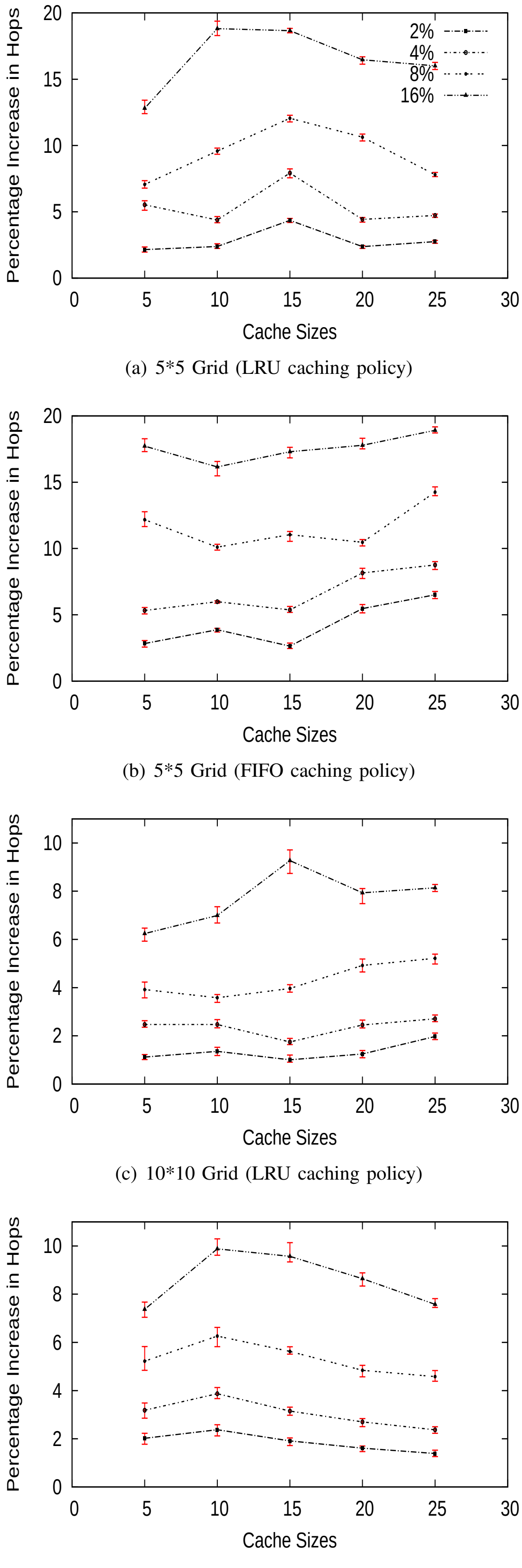
IV. Simulation

A. Simulation Setup

To exhaustively explore the scalability and effectiveness of the attack, we design our own simulator in Java. We test our attack on a 5*5 grid (small network), a 10*10 grid (medium-sized network) and two large Internet scale peer-to-peer Gnutella networks. Gnutella 6K network consists of 6301 nodes and 22077 edges while the Gnutella 8K network consists of 8846 nodes and 31839 edges. For our implementation, we consider that the Gnutella network graphs were undirected. More details about the Gnutella networks are available in [5]. The purpose of choosing these networks is to consider realistic topologies and to stress test our experiments. In fact, the Gnutella networks took several days to execute and simulate the attack.

For our simulations we consider an ICN, where every node in the network is storage-enabled. In our experiments, we assume that all nodes have the same cache size. For the grid networks, 20% of all nodes are chosen to be custodians while for the Gnutella networks, 5% of all nodes are chosen to be custodians (the Gnutella networks have thousands of nodes). The remaining nodes in the network act as requesters. Each request is generated from a Zipfian distribution and is distributed uniformly at random among the requesters. The content universe is assumed to be 10 times the number of nodes for the grid networks (i.e., 250 and 1000 for the 5*5 grid and 10*10 grid respectively) and 2000 for the large Gnutella networks. The content is distributed uniformly at random among the different custodians.

The attack is implemented in the following manner. Let us consider that a node (say A) turns into an attacker. In this scenario, A sends attack requests along with its legitimate requests. For example, if A is sending attack requests at twice



(a) 5*5 Grid (LRU caching policy)

(b) 5*5 Grid (FIFO caching policy)

(c) 10*10 Grid (LRU caching policy)

(d) 10*10 Grid (FIFO caching policy)

Figure 1. Grid Topology: Percentage Increase in Hop Count vs. Cache Size

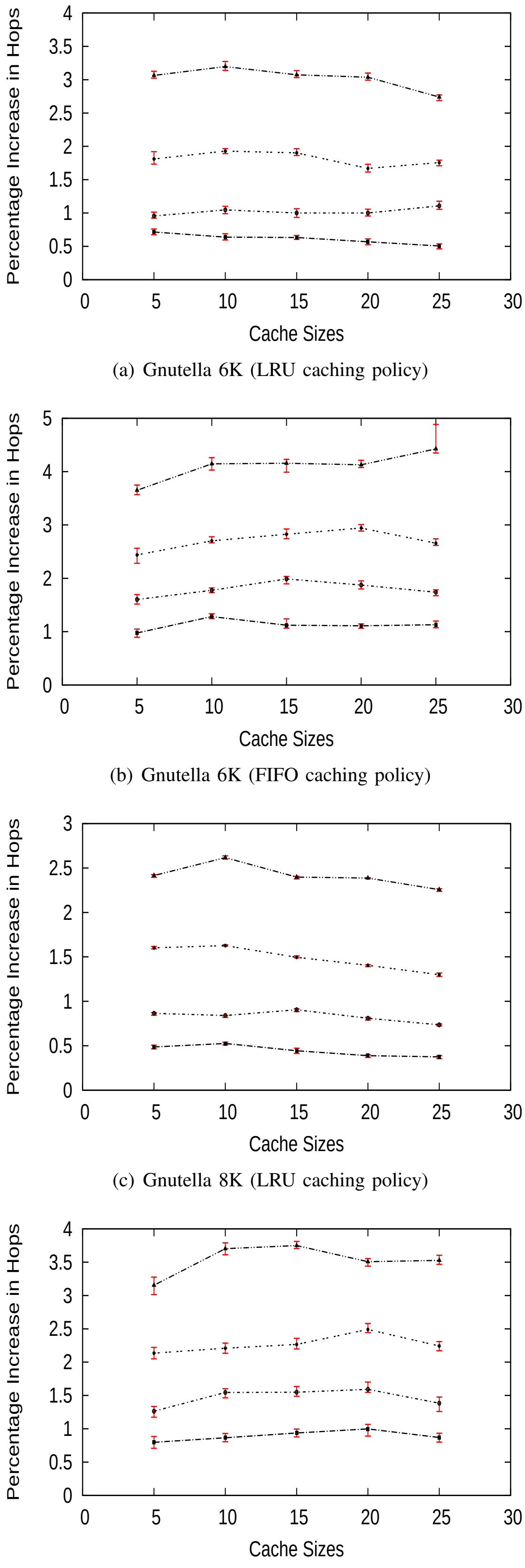
the rate of legitimate requests (denoted as attack rate of 2), then each time it sends a legitimate request, it sends two attack requests. The content for which these attack requests are issued is chosen randomly from the set of unpopular content (this set includes the least popular content from the Zipfian distribution). The set of unpopular content is chosen to be 120% of the cache size (i.e., if the cache size is 10, the set of unpopular content considered is 12).

In our simulations, the first 70% of all requests are used to warm up the caches. The attack begins only after caches have been warmed up, and we report all results on the last 30% requests. Each simulation run consists of 100000 requests with results being computed on the last 30000 requests. To ensure that our comparison is unbiased, we assume that the experimental setup remains unchanged when we are testing with and without attackers for a particular cache management policy (e.g., LRU) and cache size (e.g., 25). More precisely, we ensure that the custodian locations, the sources and the same order of legitimate requests is considered when testing with and without attackers. When attackers are added, we insert additional attack requests within the legitimate request pattern. This experimental methodology ensures a fair comparison by ensuring that custodian location and request pattern do not impact the results. Each data point on the figures is obtained as a mean of ten such runs with the errors bars denoting the difference between the maximum and minimum values. The small height of the error bars is a measure of the confidence of our results. Note that without following the above outlined methodology, we observe that the performance of the attack is worse.

B. *Results*

In our simulations we experiment with varying number of attackers (namely, 2%, 4%, 8% and 16% nodes as attackers). We experiment with multiple cache management policies (LRU, FIFO and Random) and varying values of α (0.35, 0.65, and 0.85) and with different attack rates (1, 2 and 4). We plot the average percentage increase in hop count for requests for a network under the DoS attack for varying cache sizes. Hop count is directly related to user level parameters such as delay and throughput. Note that we consider the increase in hop count only for those requests that experience a cache hit at an enroute cache when there are no attackers. This is because the DoS attack cannot increase the hop count for those requests which are served by the custodian itself during the normal operation. If we consider all requests, the attack is ineffective as the increase in hop count is small. *Overall our results show that the attack is moderately successful in small and medium scale networks and is ineffective for large networks. This conclusion holds true for a wide range of network parameters.* As the legends in our plots are repetitive, we show it only once in Fig. 1(a). Due to lack of space we mainly present results for α = 0.65 and rate =1.

Small and medium scale networks: Fig. 1 shows the percentage increase in average hop count for the LRU and FIFO caching policy with different number of attackers for small scale (5*5 grid) and medium scale (10*10 grid) networks respectively. We observe from this figure that the attack is moderately successful with percentage increase in average hop count increasing as the number of attackers increases. Even



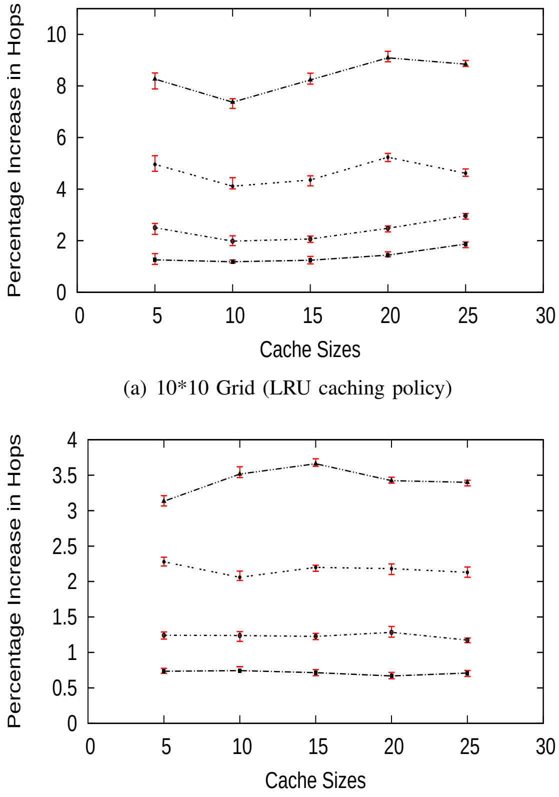
(a) Gnutella 6K (LRU caching policy)

(b) Gnutella 6K (FIFO caching policy)

(c) Gnutella 8K (LRU caching policy)

(d) Gnutella 8K (FIFO caching policy)

Figure 2. Gnutella: Percentage Increase in Hop Count vs. Cache Size



(a) 10*10 Grid (LRU caching policy)

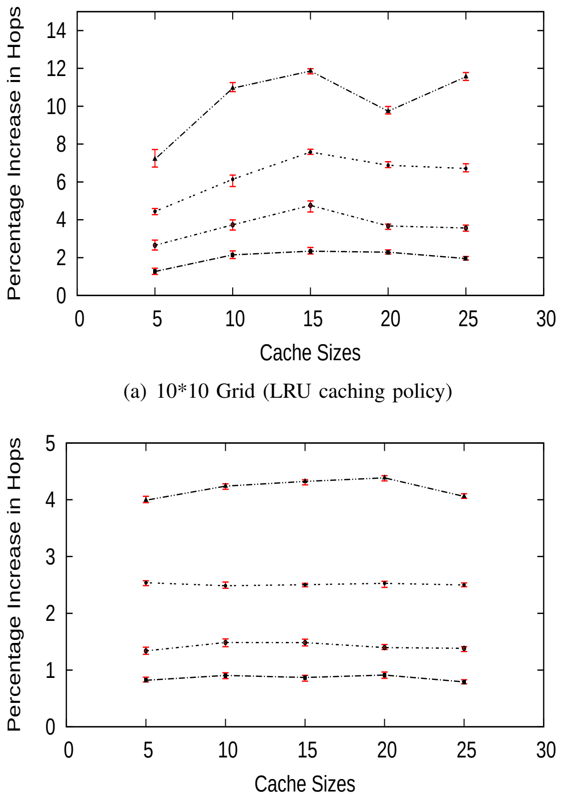
(b) Gnutella 6K (LRU caching policy)

Figure 3. Zipfian $\alpha = 0.85$: Percentage Increase in Hop Count vs. Cache Size

with a increased number of attackers, we observe that there is limited increase in the hop count. More interestingly, we observe that the attack is more potent (with increase in hop count being higher) for the FIFO caching policy in comparison to LRU. The reason for lower percentage increase in hop count for the LRU caching policy is that it evicts content based on popularity in contrast to FIFO, which evicts content based on the order of arrivals. As a result it is easier to evict popular content from a FIFO cache in comparison to LRU, which leads to higher hop count.

Large scale Gnutella networks: Fig.2 shows the percentage increase in average hop count for Gnutella 6K and the Gnutella 8K topology with different number of attackers for the LRU and FIFO caching policies. We observe that the effectiveness of the attack drastically diminishes as the network size increases and the attack becomes practically ineffective in large networks. Even with 16% nodes in the network as attackers, we observe that the percentage increase in hop count is approximately 3%. The increase in hop count for the larger Gnutella 8K topology is lower than the Gnutella 6K topology. We note that the Gnutella topologies take multiple days to execute, but they are imperative for understanding the scalability of the attack. They demonstrate that the DoS attack is unlikely to scale to large Internet scale networks with thousands of nodes.

Exploration of simulation parameters: We conduct detailed experimentation by varying the various simulation parameters. We present results for the 10*10 grid and the Gnutella 6K



(a) 10*10 Grid (LRU caching policy)

(b) Gnutella 6K (LRU caching policy)

Figure 4. Zipfian $\alpha = 0.65$, rate = 2: Percentage Increase in Hop Count vs. Cache Size

topology for the LRU caching policy for $\alpha = 0.85$ and rate = 2 (Figs. 3 and 4). Once again, we observe that the attack potency decreases with increasing network size. We observe similar results with other values of α and higher rate values. Note that high rate values are not useful, as they increase the network traffic considerably and therefore these attacks can be easily detected. We observed similar results with the Random caching policy.

Overall, our experiments yield some surprising results. We observe that the DoS attack is moderately successful in small networks. However, this DoS attack does not scale. As the size of the network increases the impact of the attack diminishes with the attack producing minimal performance degradation for large Internet scale topologies.

V. Conclusion and Future Work

In this paper, we explored the scalability and effectiveness of a cache pollution based denial of service (DoS) attack in information centric networks. Via exhaustive simulations on realistic Internet scale networks, we showed that the while the attack might be moderately successful in small networks, its potency decreased rapidly and it produced negligible impact in large networks. Our results also demonstrated that the attack is likely to be more successful against the FIFO caching policy in comparison to LRU.

In future, we plan to explore avenues for improving this attack. One approach to increase the potency of a cache pollution based DoS attack might be to space requests for same content in time intelligently. A concept of wait time can be introduced, where an attacker waits a specific amount of time before requesting the same content again. The rationale behind this approach is that once a content enters a cache, it is likely to remain there for certain duration of time in future before getting evicted. Requesting the same content during that time window again will increase the attack rate, without increasing the potency of the attack. A possible approach to estimate the wait time might be to use the concept of characteristic time [15]; characteristic time for a cache indicates the expected amount of time a requested content is likely to remain in the cache before getting evicted. Our initial exploration with this concept of wait time has yielded interesting results and appears to improve the attack potency. We plan to explore this attack in detail in future.

References

[1] M. Conti, P. Gasti, and M. Teoli, "A lightweight mechanism for detection of cache pollution attacks in named data networking," *Computer Networking*, vol. 57, no. 16, pp. 3178–3191, 2013.

[2] "Project ccnx," https://www.ccnx.org/.

[3] W. K. Chai, D. He, I. Psaras, and G. Pavlou, "Cache "less for more" in information-centric networks," in *Proceedings of the 11th International IFIP TC 6 Conference on Networking - Volume Part I*, ser. IFIP'12, 2012, pp. 27–40.

[4] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, and S. Nanda, "Congestion-aware caching and search in information-centric networks," in *Proceedings of the 1st International Conference on Information-centric Networking*, ser. ICN '14, 2014, pp. 37–46.

[5] "Stanford network analysis project (snap)," http://snap.stanford.edu/.

[6] E. AbdAllah, H. Hassanein, and M. Zulkernine, "A survey of security attacks in information-centric networking," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, 2015.

[7] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang, "Dos and ddos in named data networking," in *22nd International Conference on Computer Communications and Networks (ICCCN), 2013*, July 2013, pp. 1–7.

[8] A. Afanasyev, P. Mahadevan, I. Moiseenko, E. Uzun, and L. Zhang, "Interest flooding attack and countermeasures in named data networking," in *IFIP Networking Conference, 2013*, May 2013, pp. 1–9.

[9] M. Xie, I. Widjaja, and H. Wang, "Enhancing cache robustness for content-centric networking," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 2426–2434.

[10] A. Karami, "The use of computational intelligence for security in named data networking," Ph.D. dissertation, Universitat Politecnica de Catalunya Barcelona, 2015.

[11] A. Mohaisen, X. Zhang, M. Schuchard, H. Xie, and Y. Kim, "Protecting access privacy of cached contents in information centric networks," in *Proceedings of the ACM Conference on Computer and Communications Security*, ser. CCS '13, 2013.

[12] S. Misra, R. Tourani, and N. E. Majd, "Secure content delivery in information-centric networks: Design, implementation, and analyses," in *Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '13, 2013, pp. 73–78.

[13] W. Wong and P. Nikander, "Secure naming in information-centric networks," in *Proceedings of the Re-Architecting the Internet Workshop*, ser. ReARCH '10, 2010, pp. 12:1–12:6.

[14] A. Mohaisen, H. Mekky, X. Zhang, H. Xie, and Y. Kim, "Timing attacks on access privacy in information centric networks and countermeasures," *Dependable and Secure Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[15] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, 2006.