Avoiding Notorious Content Sources: A Content-Poisoning Attack Mitigation Approach

1st Ioanna Angeliki Kapetanidou
Democritus University of Thrace
Xanthi, Greece
ikapetan(at)ee.duth.gr

2nd Stavros Malagaris
Democritus University of Thrace
Xanthi, Greece
stavmala(at)ee.duth.gr

3rd Vassilis Tsaoussidis
Democritus University of Thrace
Xanthi, Greece
vtsaousi(at)ee.duth.gr

***Abstract*—Named Data Networking (NDN) has emerged as a promising Future Internet architecture. NDN provisions security by design and guarantees that data packets are immutable and authentic. Nevertheless, its inherent in-network caching feature has opened the door to new types of security attacks. One such critical security issue in NDN is content poisoning attacks. In content poisoning, the attacker aims at injecting poisonous (i.e., fake or invalid) content in the network caches. In this paper, we propose a reputation-based content poisoning mitigation model, which assists both the access and the core network nodes in identifying the sources from which poisonous content is originated, and subsequently, limiting the Interest flow towards those notorious sources as well as in avoiding caching poisonous content.**

***Index Terms*—Information-Centric Networking, Named Data Networking, Attack, Security, Reputation**

I. Introduction

Naming the content itself, as opposed to addressing the respective end-hosts, constitutes the fundamental concept of the Named Data Networking (NDN) [1] architecture, a prominent Information Centric Networking (ICN) paradigm. More specifically, NDN content-consumers issue Interest Packets denoting the desired content name. NDN networks respond to the consumer's Interests by returning matching Data Packets.

Named content allows NDN for supporting two interrelated key features, i.e. in-network caching and data-centric security. The former is achieved by requiring routers to maintain a so-called 'Content Store' (CS), i.e., a temporary cache. Copies of the content objects can be cached in the intermediate nodes as the content traverses the network and be used later to satisfy subsequent requests for the same content. Therefore, in-network caching contributes to minimizing the delivery latency and the bandwidth requirements.

Regarding security, NDN mandates content producers to sign data upon its creation using public key cryptography. Consumers are also obliged to verify the signatures of the received Data Packets, so that they distinguish and discard the invalid content. This way, asynchronous data creation and consumption is doable, thanks to the in-network caching, while, at the same time, data integrity and originality are ensured.

Nevertheless, the in-network caching, besides its obvious benefits, has also revealed new security threats in NDN; content poisoning attacks is such a crucial threat. More specifically, routers, unlike consumers, do not verify the incoming content's signature because the introduced computational cost for performing in-network cryptographic operations is considered prohibitive. As a result, content is inserted into the routers' caches without any preceding validation.

Building upon this vulnerability, in content poisoning, the adversary aims at flooding the network with bogus content. A recent study [2] has confirmed the feasibility of launching content poisoning attacks on real NDN networks. It also demonstrated the attack's impact on the NDN nodes, including both the edge and core routers, apart from the clients and content providers.

In this paper, we are concerned with mitigating content poisoning attacks, especially in circumstances, when the same content can be provided by different content-sources, each one corresponding to a different namespace. In this context, we adopt reputation as a pertinent way to complement the NDN security functions [3], and propose a reputation-based system aiming to identify namespaces under attack, and hence, prevent content created under such namespaces from being spread.

To this end, our solution is premised on two NDN native functionalities: Interest forwarding and in-network caching. On the one hand, we propose a 'Forwarding Adaptation' approach at the edge on which the Interests destined for notorious namespaces are being limited, while, on the other hand, we propose a 'Caching Adaptation' approach, which assists network routers in steering clear of caching poisonous content. We also evaluate the joint operation of the two proposed approaches.

The contribution of our work lies in the following aspects:

- To the best of our knowledge, our solution is the first one to consider potential network overloading by Interests corresponding to poisonous content, and subsequently, enforce restrictions. As yet, controlling the Interest flow has been leveraged mainly to mitigate another type of attacks in NDN, i.e. Interest Flooding Attacks (IFAs) [1] [4]–[6].
- Our solution is versatile in the sense that it can be integrated either at the access or at the core network,

[1]In IFAs, the attacker depletes the network resources with Interests for non-existent content

978-1-6654-9792-3/22/$31.00 ©2022 IEEE

or even at both sides, depending on the attack scenario constraints.
- We propose a lightweight solution, which does not require executing any cryptographic operations or sending reports to routers for every received Data Packet.

The rest of the paper is organized as follows: In Section II, we provide a background on NDN and content poisoning attacks, as well as we overview related works. In Sections III and IV, we detail the design and present the evaluation of the investigated mitigation approaches, respectively. We summarize our concluding remarks in Section V.

II. Background and Related Work

A. Named Data Networking

NDN's fundamental principle is naming directly the content objects. In other words, content becomes directly addressable and routing is based on content names. NDN names follow hierarchical structure (e.g. /duth.gr/eece/media/presentation.mp4), in which the prefix usually denotes the content producer namespace.

NDN adheres to a request/response communication model, which functions by using two types of packets: Interest packets and Data Packets. In particular, a consumer who wants to obtain specific content, forwards an Interest Packet to the network, which specifies the name of the desired content. As a response to the consumer's Interest, a matching Data Packet with the desired content, fetched either by the content source or an on-path router, is delivered.

Moreover, NDN requires in-network caching by routers at their 'Content Stores' (CS). In addition, every NDN node also maintains a 'Pending Interest Table' (PIT) and a 'Forwarding Information Base' (FIB). In PIT closely spaced pending interests for the same content are being aggregated in a common entry including the interested interfaces, while FIB is similar to a routing table, providing information regarding forwarding outgoing Interests towards the initial content producer.

Security also follows a data-centric approach in NDN [7]. More specifically, NDN provides secured Data Packets by requiring content publishers to seal produced data with public key signatures, but also, consumers to verify signatures. Furthermore, NDN utilizes digital certificates [8] and 'trust schemas' [9] to specify under which namespaces each key is authorized to sign created content.

B. Content Poisoning Attacks

As performing signature verification constitutes a deterrent policy for routers [10], the latter are left susceptible to content poisoning attacks.

1) Attack Model: In a content poisoning attack, the adversary attempts to fill the routers' CSs with either fake (generated by an incorrect producer) or corrupted (i.e. carrying an invalid signature) content. To launch a content poisoning attack, the adversary anticipates genuine interests for a valid content name and subsequently, injects poisonous content in the network/caches. In particular, while the poisonous content travels back to the Interest issuer, it is cached by the routers along the path. To make things worse, subsequent Interests for the same content are satisfied by the infected CS and hence, the poisonous content is spread to the network. This results in increased content delivery cost for both the consumers and the network [11], and in decreased QoS for end-users.

2) Mitigation Efforts: Various mitigation mechanisms have been proposed. First off, the Interest-Key Binding (IKB) rule [10] proposes a light-weight verification approach by exploiting the PublisherPublicKeyDigest (PPKD) Interest field which contains the SHA-256 digest of the publisher's public key. Routers compare the received Data packet's public key hash to the PPKD of the corresponding PIT entry, and, subsequently, cache and forward the Data Packet iff those two match one another. Nonetheless, the overhead and computational cost remain high given that consumers need to pre-fetch the producer's PPKD and routers to verify PPKD on every hop.

Hashing is also utilized in [12] which stipulates that consumers receiving invalid content notify the initial content publisher who, in turn, broadcasts a message to urge network routers to exclude the content. Similar to [12], authors in [13] also propose a lightweight verification operation in the context of NDN-based IoT networks, and notify the routers in case of a negative result. However, the aforementioned techniques lie mainly in proposing a lightweight content integrity checking by consumers, which is an NDN inherent characteristic.

Aiming to reduce the verification cost, [14] probabilistically verifies and caches the validated popular content. However, attackers can launch a verification attack [15], i.e. repeatedly request specific content to enforce routers to perform redundant verification operations and consequently, increase the network latency. Probabilistic in-network verification is also used by the In-network Collaborative Verification (ICoV) approach [16], in case the router is overloaded. The probability is based on the incoming interface's credibility value.

Further efforts aiming to make the verification process more lightweight include symmetric encryption, such as in the Router-Cooperation scheme [17], or self-certifying packet names, e.g. [18], [19], which also imply additional overhead.

Moreover, [20], [21] and [22] rely on explicit exclusion filters (i.e., by using the optional "Exclude" field of Interest packets) to allow consumers for issuing Interests excluding invalid content. However, the exclusion functionality is no longer supported in NDN[2], nullifying the previous methods.

'Lossy Caching' [23], [24] allows certain data to be cached on the grounds that although there is a tradeoff between the cache hit ratio and restricting caching to a subset of verified content, the network performance is only mildly affected.

Some forwarding-based mitigation methods have also been proposed, i.e. Feedback-based Content Poisoning Mitigation (FCPM) [25], 'Immediate Failover' and 'Probe First' [26], Router-Oriented Mitigation (ROM) [27] and Ant Colony Algorithm Based Content Poisoning Mitigation (ACO-CPM) [28]. 'Immediate Failover' makes the next hop that returned invalid content the least preferred forwarding option for subsequent

[2]since version 0.3

Interests, whereas 'Probe First' stops forwarding to such next hops temporarily, replays prior Interests and restarts forwarding once the received data is verified. The FCMP blocks malicious content sources and discards the poisonous content from the CS based on the consumer's feedback. ROM and ACO-CPM are premised on the idea that on-path malicious routers might also disseminate poisonous content. In ROM, each router assess the trustworthiness of its adjacent routers, based on the number of received poisonous packets reported by consumers and decides whether to include each neighbor in the forwarding path, accordingly.

However, the aforementioned strategies necessitate routers to retain the history of old Interests and the next-hop that replied to those Interests with invalid content and consumers to send one report per received data packet to the routers, thus entailing extra overhead.

ACO-CPM introduces 'Interest Ant' and 'Data Ant' packets which are used as probes to periodically evaluate the credibility of on-path routers and rank the potential forwarding interfaces accordingly, aiming at a secure forwarding path for the next Interests. Interest Ants also carry the hash value of poisonous content, so that polluted routers can be notified to discard it from their caches.

Given the recent advances described above, we argue that our work is the first one to offer the flexibility in safeguarding the NDN network against an ongoing content poisoning attack either at the access or at the core network-side, by controlling the forwarding or the caching respectively, or even leverage those two options jointly to eliminate content poisoning.

III. Design

In this section, we set out the design primitives of our mechanism as well as the specifications of each of the two mitigation methods.

It should be noted that we assume that content pieces can be found at different sources of a pool of namespaces and each namespace has a distinctive prefix.

A. Reputation-based Attack Identification

The fundamental goal of our system is to be able to detect a content source under attack and in turn, to temper the attack by avoiding the interactions with the respective namespace.

To this end, we introduce the *'Notoriety'* metric, which is defined as the ratio of the number of the poisonous received content objects over the total received content objects and is calculated using the following equation:

$$Notoriety = \frac{No.\ Poisonous\ Received\ Content\ Objects}{No.\ Total\ Received\ Content\ Objects} \tag{1}$$

Therefore, the Notoriety value indicates the proportion of the poisonous content.

Notoriety is computed for each namespace and is refreshed upon every receipt of a content piece. Apparently, the Notoriety values are inferred by the consumers who are capable of distinguishing the invalid content via signature verification.

The two below-described mitigation approaches make appropriate decisions based on the Notoriety values.

B. Forwarding Adaptation

In the first mitigation model, a 'Forwarding Probability' (FP) tied to the Notoriety value is introduced. The FP is calculated as:

$$FP = 100 - Notoriety \tag{2}$$

and hence, the higher the Notoriety value, the lower the Forwarding Probability. The FP can be either computed at the consumer end or by its adjacent access routers, in order to be leveraged for limiting the flow of Interests requiring potentially poisonous content.

On every Interest issuing, the consumer (or the edge router) generates a random number (r) which needs to be greater than the specific namespace's FP in order to forward the Interest to the core network. In the reverse case, the Interest is dropped and the procedure is repeated for the next selected namespace, as indicated in the following formula:

$$Forwarding\ Decision = \begin{cases} Issue\ Interest, & \text{if } r > FP. \\ Drop\ Interest, & \text{otherwise.} \end{cases} \tag{3}$$

This method behaves proactively as it limits the flow of poisonous content by avoiding forwarding Interests for notorious namespaces in the first place. Therefore, the forwarding adaptation provides protection against content poisoning indirectly, by influencing the chances of a malicious source to receive an Interest and reply with poisonous content that can be cached.

Moreover, adapting the Interest forwarding also saves time and reduces the computational cost for the consumers who would otherwise keep requesting notorious content that would eventually discard after finding out that its signature cannot be verified.

Along the lines of the 'Intermediate Failover' strategy described in [26], notorious namespaces are not explicitly excluded. However, unlike 'Intermediate Failover' which prioritizes the legitimate forwarding options over those of ill repute, our approach is more lenient in the sense that by utilizing the probabilistic forwarding, notorious namespaces are even offered the chance to satisfy some Interests occasionally, despite of their relative ranking, such that they can re-gain the consumers' trust upon potential recovery.

C. Caching Adaptation

The second approach aims to shield the on-path routers' caches, i.e, their weak point. In this case, the routers need to monitor the content that pass through them and keep track of the related namespaces and periodically communicate with the consumers to fetch the associated Notoriety values.

It is important to note that, in our approach, contrary to previous works, using reports regarding each distinct content object, routers fetch one report per namespace including the calculated Notoriety value. This happens periodically so that routers become aware of the latest Notoriety values.

Similar to the 'Forwarding Adaptation', each router defines a 'Cache Probability' (CP) for each prefix which increases as the Notoriety value decreases:

$$CP = 100 - Notoriety \tag{4}$$

Moreover, this time the router computes a random value r, which is being compared to the CP and the caching decision is made accordingly as declared in the formula below:

$$Caching\ Decision = \begin{cases} Cache\ Content, & \text{if } r > CP. \\ Discard\ Content, & \text{otherwise.} \end{cases} \tag{5}$$

IV. Evaluation

A. Simulation Setup

To evaluate the two mitigation approaches, as well as their combined operation, we used the ndnSIM simulator [29].

For our simulations, we considered a simple tree-like topology as depicted in Figure 1. We assume that the leaf nodes are the content providers, of which the second leftmost producer has been compromised. It should be noted that the attacker responds to several Interests under its namespace with invalid content, but also replies with valid content sometimes, aiming to confuse the reputation model. The consumer (attached to the root/edge router) generates Interests for random prefixes.

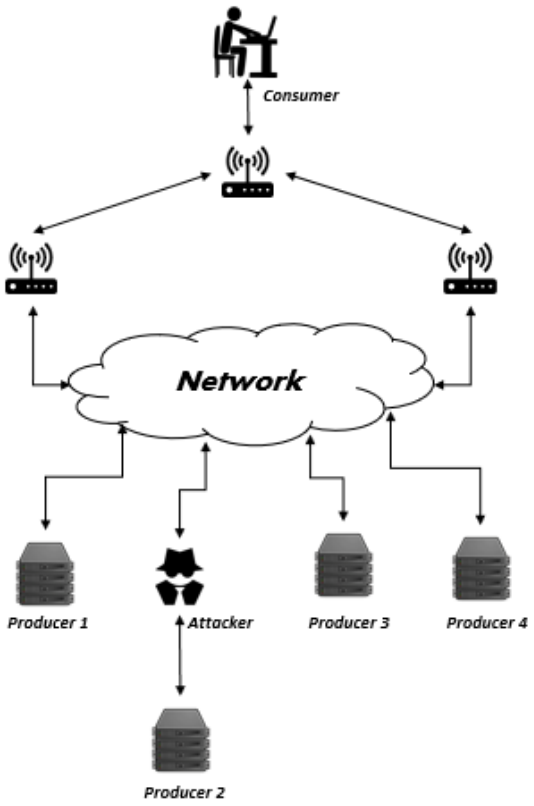

Fig. 1. Topology

TABLE I
Simulation Configurations

Parameter	Value
Link Bandwidth	10 Mbps
Link delay	10 ms
Default Content Store size	800 packets
Cache replacement policy	LRU
Forwarding Strategy	Multicast

Further configurations are included in Table I.

The initial Notoriety value of all namespaces is considered to be zero. In addition, the mitigation mechanism is being activated once the content stores have been filled up.

B. Results

We have performed two types of simulations, each time investigating different aspects. More specifically, in the first experiment we evaluate the efficiency of the mitigation approaches, while in the second one we evaluate the reaction time of each countermeasure.

Since the results are dependent on non-deterministic aspects (such as the distribution of interests per namespace over time, the number of cached poisonous packets on the time of the mitigation mechanism activation and the computed probabilities), we have repeated each simulation 10 times, to ensure their reliability.

1) Simulation Type 1: Efficiency Evaluation: In the first type of simulations, we evaluate the impact of the content poisoning attack both consumer- and router-wise under all three mitigation approaches, by considering the following evaluation metrics:

- Number of Forwarded Interests per Namespace
- Percentage of Forwarded Interests for the Notorious Namespace
- Number of Cached Data Packets per Namespace
- Percentage of Poisonous Cached Data Packets

The simulation results are presented in Figures 2-5. We compare our results against the case of an attack with no mitigation method in force.

Figure 2 depicts the number of the forwarded Interests per namespace in each case, while Figure 4 illustrates the number of the Data Packets cached at the edge router for the four examined cases. In Figures 2 and 4, the red box corresponds to the notorious namespace and the green ones to the legitimate namespaces. The orange and the blue lines are the meanlines, i.e. indicate the average values, for the notorious and the legitimate namespaces, respectively.

Figures 3 and 5 depict the percentage of the Interests for the notorious namespace that have been eventually forwarded and the percentage of the cached data packets corresponding to poisonous content, respectively. The results are averaged over 10 runs.

Not very surprisingly, the combined implementation of the two approaches produces the best results. This is obvious since that is the only case in which not only the Interests destined

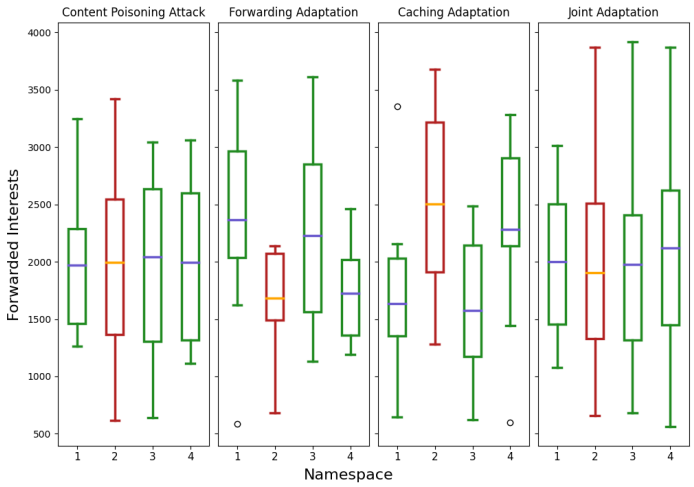

Fig. 2. No. Forwarded Interests

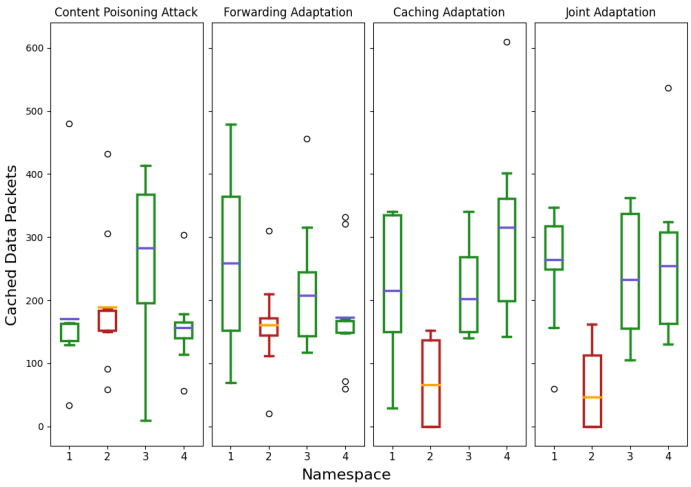

Fig. 4. No. Cached Data Packets

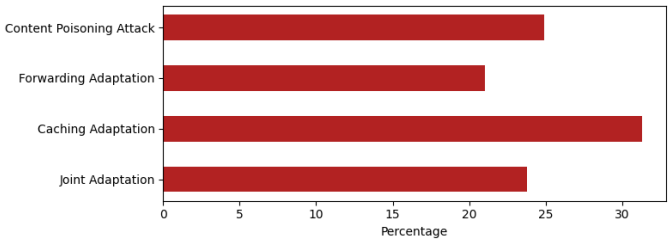

Fig. 3. Percentage of Interests for the Notorious Namespace Forwarded

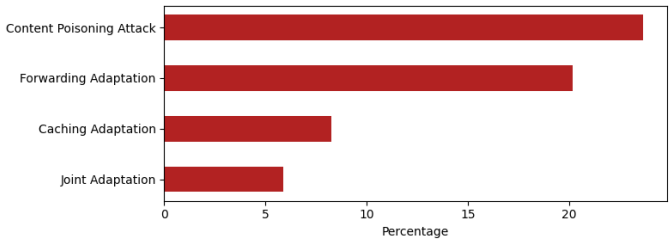

Fig. 5. Percentage of Poisonous Cached Data Packets

for the notorious namespace are less compared to any of the legitimate namespaces, but also the cached content originating by the notorious namespace has been reduced more than in any other case.

Such results were expected as in the joint adaptation case, protection against content poisoning is provided both proactively on the consumer-side and further reinforced by the core network routers. Nevertheless, even when each method is implemented separately, it achieves sufficient attack mitigation, despite the adversary's attempts to slip through the mitigation mechanism by behaving legitimately from time to time.

Although handling directly the caching is obviously more efficient (reducing the poisonous content found in the CSs by around 65% in 'Caching Adaptation' and by 75% in the 'Joint Adaptation'), even by simply controlling the Interest forwarding, the poisonous content worming itself into the network caches is approximately 15% less than the respective in the case of no mitigation.

2) Simulation Type 2: Reaction Time Evaluation: In the second type of simulations, we assess the reaction time needed for each approach to reach a specific threshold value at which the CS is considered uncontaminated.

As shown in Figure 6, we experiment with three CS sizes: the default one (800 packets), half the default CS size (400 packets) and double the default CS size (1600 packets). We also set the threshold values equal to 5%, 10% and 20%.

The x-axis labels in the subplots correspond to the three different mitigation approaches. In particular, FA stands for the 'Forwarding Adaptation', CA stands for the 'Caching Adaptation' and Joint stands for the 'Joint Adaptation'. The orange lines in the boxes are the meanlines.

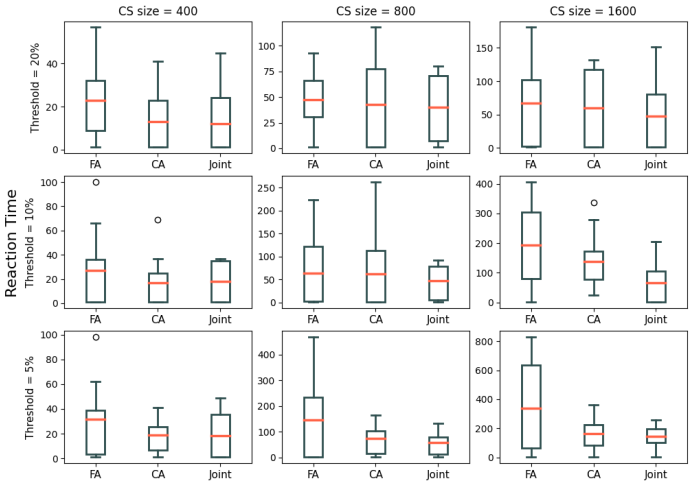

Fig. 6. Reaction time for threshold values set to 5%, 10% and 20%, and CS sizes equal to 400, 800 and 1600 packets

As expected, as the cache size increases, the time it takes for each adaptation approach to reach the specified threshold

increases too. However, the 'Forwarding Adaptation' needs the most time to flush the poisonous content from the routers' caches, even in the case of the highest threshold value (20%) and the lower CS size (400 packets). This becomes even more apparent when the threshold set to 5%. In this case, the 'Forwarding Adaptation' requires 50% more time than the 'Caching Adaptation' approach and almost 57% more time than the 'Joint Adaptation' approach, on average.

V. Conclusions and Future Work

The current NDN security plane is inherently susceptible to content poisoning attacks. In this work, we have developed a reputation-based system which mitigates such attacks in NDN networks by exploiting two NDN intristic features: the forwarding at the edge and the caching at the core network. We demonstrated that by exploiting those two features to avoid the notorious sources, the content poisoning attack impact is reduced significantly.

In the future, we desire to compare our approach against specific forwarding-based mitigation strategies. In addition, we consider extending our mechanism to address diverse use-cases besides content poisoning attacks, e.g. computing the Notoriety values based on criteria other than content validity. Lastly, we aspire to integrate our model with a distributed reputation system, to allow for Notoriety values to be loaded to and fetched by a tamper-proof distributed ledger.

References

[1] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[2] T. Nguyen, X. Marchal, G. Doyen, T. Cholez, and R. Cogranne, "Content poisoning in named data networking: Comprehensive characterization of real deployment," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 72–80.

[3] I. A. Kapetanidou, C.-A. Sarros, and V. Tsaoussidis, "Reputation-based trust approaches in named data networking," *Future Internet*, vol. 11, no. 11, p. 241, 2019.

[4] S. Umeda, T. Kamimoto, Y. Ohata, and H. Shigeno, "Interest flow control method based on user reputation and content name prefixes in named data networking," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1. IEEE, 2015, pp. 710–717.

[5] A. Compagno, M. Conti, P. Gasti, and G. Tsudik, "Poseidon: Mitigating interest flooding ddos attacks in named data networking," in *38th annual IEEE conference on local computer networks*. IEEE, 2013, pp. 630–638.

[6] A. Benarfa, M. Hassan, A. Compagno, E. Losiouk, M. B. Yagoubi, and M. Conti, "Chokifa: A new detection and mitigation approach against interest flooding attacks in ndn," in *International Conference on Wired/Wireless Internet Communication*. Springer, 2019, pp. 53–65.

[7] Z. Zhang, Y. Yu, H. Zhang, E. Newberry, S. Mastorakis, Y. Li, A. Afanasyev, and L. Zhang, "An overview of security support in named data networking," *IEEE Communications Magazine*, vol. 56, no. 11, pp. 62–68, 2018.

[8] Z. Zhang, A. Afanasyev, and L. Zhang, "Ndncert: universal usable trust management for ndn," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 2017, pp. 178–179.

[9] Y. Yu, A. Afanasyev, D. Clark, V. Jacobson, L. Zhang *et al.*, "Schematizing trust in named data networking," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*. ACM, 2015, pp. 177–186.

[10] C. Ghali, G. Tsudik, and E. Uzun, "In content we trust: Network-layer trust in content-centric networking," *IEEE/ACM Transactions on Networking*, vol. 27, no. 5, pp. 1787–1800, 2019.

[11] P. Gasti and G. Tsudik, "Content-centric and named-data networking security: The good, the bad and the rest," in *2018 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. IEEE, 2018, pp. 1–6.

[12] S. S. Ullah, I. Ullah, H. Khattak, M. A. Khan, M. Adnan, S. Hussain, N. U. Amin, and M. A. K. Khattak, "A lightweight identity-based signature scheme for mitigation of content poisoning attack in named data networking with internet of things," *IEEE Access*, vol. 8, pp. 98 910–98 928, 2020.

[13] S. Hussain, S. S. Ullah, A. Gumaei, M. Al-Rakhami, I. Ahmad, and S. M. Arif, "A novel efficient certificateless signature scheme for the prevention of content poisoning attack in named data networking-based internet of things," *IEEE Access*, vol. 9, pp. 40 198–40 215, 2021.

[14] D. Kim, S. Nam, J. Bi, and I. Yeom, "Efficient content verification in named data networking," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, 2015, pp. 109–116.

[15] D. Kim, J. Bi, A. V. Vasilakos, and I. Yeom, "Security of cached content in ndn," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 12, pp. 2933–2944, 2017.

[16] H. Kang, Y. Zhu, Y. Tao, and J. Yang, "An in-network collaborative verification mechanism for defending content poisoning in named data networking," in *2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*. IEEE, 2018, pp. 46–50.

[17] Y. Wang, Z. Qi, K. Lei, B. Liu, and C. Tian, "Preventing" bad" content dispersal in named data networking," in *Proceedings of the ACM Turing 50th Celebration Conference-China*, 2017, pp. 1–8.

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

[19] M. Baugher, B. Davie, A. Narayanan, and D. Oran, "Self-verifying names for read-only named data," in *2012 Proceedings IEEE INFOCOM Workshops*, 2012, pp. 274–279.

[20] C. Ghali, G. Tsudik, and E. Uzun, "Needle in a haystack: Mitigating content poisoning in named-data networking," in *Proceedings of NDSS Workshop on Security of Emerging Networking Technologies (SENT)*, 2014.

[21] Z. Rezaeifar, J. Wang, and H. Oh, "A trust-based method for mitigating cache poisoning in name data networking," *Journal of Network and Computer Applications*, vol. 104, pp. 117–132, 2018.

[22] X. Hu, J. Gong, G. Cheng, G. Zhang, and C. Fan, "Mitigating content poisoning with name-key based forwarding and multipath forwarding based inband probe for energy management in smart cities," *IEEE Access*, vol. 6, pp. 39 692–39 704, 2018.

[23] G. Bianchi, A. Detti, A. Caponi, and N. Blefari Melazzi, "Check before storing: What is the performance price of content integrity verification in lru caching?" *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, p. 59–67, Jul. 2013. [Online]. Available: https://doi.org/10.1145/2500098.2500106

[24] A. Detti, A. Caponi, G. Tropea, G. Bianchi, and N. Blefari-Melazzi, "On the interplay among naming, content validity and caching in information centric networks," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 2108–2113.

[25] W. Cui, Y. Li, Y. Xin, and C. Liu, "Feedback-based content poisoning mitigation in named data networking," in *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2018, pp. 759–765.

[26] S. DiBenedetto and C. Papadopoulos, "Mitigating poisoned content with forwarding strategy," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 164–169.

[27] D. Wu, Z. Xu, B. Chen, and Y. Zhang, "What if routers are malicious? mitigating content poisoning attack in ndn," in *2016 IEEE Trustcom/BigDataSE/ISPA*. IEEE, 2016, pp. 481–488.

[28] W. Cui, Y. Li, Y. Zhang, C. Liu, and M. Zhan, "An ant colony algorithm based content poisoning mitigation in named data networking," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2019, pp. 176–183.

[29] S. Mastorakis, A. Afanasyev, and L. Zhang, "On the evolution of ndnsim: An open-source simulator for ndn experimentation," *ACM SIGCOMM Computer Communication Review*, vol. 47, no. 3, pp. 19–33, 2017.