Computers & Security 88 (2020) 101628

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

# InterestFence: Simple but efficient way to counter interest flooding attack☆

Jiaqing Dong[a], Kai Wang[b,*], Wei Quan[c], Hao Yin[a]

[a] Research Institute of Information Technology, Tsinghua University, Beijing, China
[b] School of Computer Science and Technology, Harbin Institute of Technology, Weihai, Shandong, China
[c] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China

ARTICLE INFO

*Article history:*
Received 19 May 2019
Revised 12 September 2019
Accepted 26 September 2019
Available online 27 September 2019

*Keywords:*
Interest flooding attack (IFA)
Named data networking (NDN)
Information centric networking (ICN)
Network security

ABSTRACT

The Interest Flooding Attack (IFA) has been one of the biggest threats to the Named Data Networking (NDN) paradigm. It is easy to launch but very difficult to mitigate. In this paper, a lightweight yet efficient IFA countermeasure, named as InterestFence, is proposed to achieve accurate detection as well as efficient attack-traffic filtering without harming any legitimate Interests. First, InterestFence detects IFAs based on the content servers rather than routers to guarantee accurate detection, since only content servers know exactly IFA's existence by checking their content index. Second, for each name prefix in every content server, all of the content items with that prefix have a hash-based security label (HSL) to claim their existence. Then an HSL verification method is securely transmitted to the involved routers to help accurately filter IFA traffic, by simply performing HSL verifying operations against malicious name prefixes. Performance evaluation demonstrates that InterestFence can filter 100% IFA traffic at intermediate routers, and keep the same level of service latency for legitimate users, while with a much lower overhead in time consumption compared with cryptographic algorithms.



## 1. Introduction

With the significant growth of Internet traffic generated by emerging types of applications, the location-based content-delivering paradigm of the traditional Internet has shown limitations. The key deviation is its attempt to build an efficient content-centric service model over a networking architecture originally designed for host-to-host conversations between remote users (Carofiglio et al., 2013; Zhang et al., 2016). To fill this gap, Named Data Networking (NDN) (Jacobson et al., 2012; NDN-NP) argues to evolve the current Internet from host-based IP networks to data-centric inter-networking paradigms, by directly placing content-distribution services at the network-layer (Mangili et al., 2016). NDN has attracted wide research attention (Xylomenos et al., 2014) since it can not only directly connect people with content and information (Kurose, 2014; Posch et al., 2017), but also facilitate future networking requirements, such as 5G (Zhang et al., 2017), Internet of Things (Hahm et al., 2017), and vehicular networks (Quan et al., 2014; Su et al., 2017).

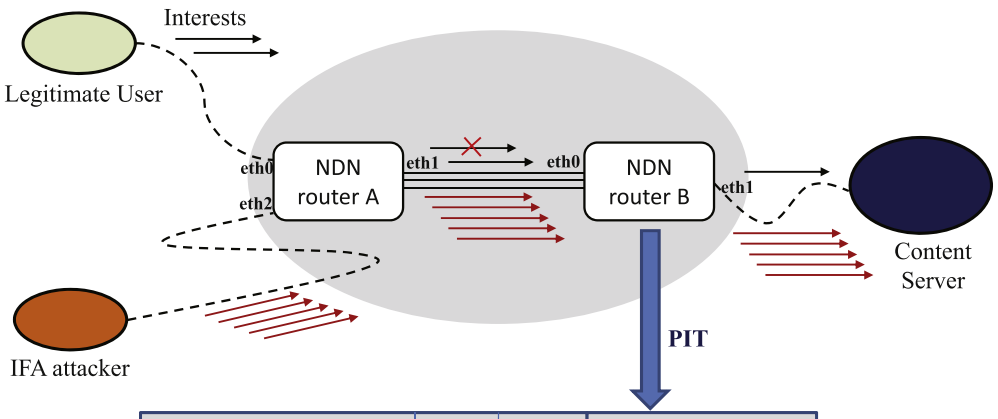
As NDN gradually develops and matures, the security concerns become increasingly critical and important. It may significantly thwart the real-world deployment of NDN if not given enough attention (Ngai et al., 2017). NDN embeds some critical security primitives in its original architecture by securing the content (Jacobson et al., 2012), and successfully reduces the impact of the notorious Distributed Denial-of-Service (DDoS) attacks (Liu et al., 2010; Zargar et al., 2013) by its receiver-driven data-retrieval model. However, its Pending Interest Table (PIT) component in each router opens up an opportunity for a new type of NDN-specific DDoS attack–the Interest Flooding Attack (IFA). In recent years, IFAs have become one of the most dangerous threats to NDN (Tourani et al., 2017).

PIT is one of the fundamental components of every NDN router. An NDN router records all of the ongoing communication states as its PIT entries, where the names as well as the incoming interfaces of each pending Interest packet are cached, until the requested data packets are returned from corresponding content servers. Under normal conditions, PIT size remains small in typical network settings, even in the absence of NDN data caching or optimal network bandwidth usage, because every pending PIT entry can be eliminated from a router's memory approximately at

☆ An earlier version of the paper (Dong et al., 2018) was presented by the 18th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2018). This version has been extended and enhanced both the key design details and performance evaluation, by at least 50% new content compared with the earlier version in the ICA3PP 2018 conference.

* Corresponding author.
*E-mail address:* dr.wangkai@hit.edu.cn (K. Wang).

https://doi.org/10.1016/j.cose.2019.101628
0167-4048/




| Interest Name | In | Out | Time since Arrival |
|---|---|---|---|
| /China/Sina/video/legi1 | eth0 | eth1 | 12 *ms* |
| /China/Sina/video/attacks1 | eth0 | eth1 | 2000 *ms* |
| /China/Sina/video/attacks2 | eth0 | eth1 | 1900 *ms* |
| /China/Sina/video/attacks3 | eth0 | eth1 | 1800 *ms* |
| /China/Sina/video/attacks4 | eth0 | eth1 | 1750 *ms* |
| /China/Sina/video/attacks5 | eth0 | eth1 | 1745 *ms* |

Malicious Interests

**Fig. 1.** Interest flooding attack.

the Round-Trip Time (RTT) scale when requested data packets return (Carofiglio et al., 2015). However, if the requested content cannot be found even in content servers located at remote edge networks, its related PIT entry would not be deleted until the Time-To-Live (TTL) of this entry expires. The timescale of TTL is much longer than that of RTT, by up to 3 orders of magnitude (Afanasyev et al., 2012; Carofiglio et al., 2011; Mastorakis et al., 2017; Wang et al., 2014a). If too many fake Interests are issued for non-existent content, they will cause a significant consumption of memory resources of each router along the forwarding path, as well as computation resources of victim content servers.

**What is an IFA:** An IFA exploits the above NDN PIT features, and aims at achieving denial of service for legitimate users by flooding excessive amount of fake Interests to exhaust critical network resources. These fake Interests can finally reach the victim content servers without any cache hit, and meanwhile the records for them can stay in the router's PIT until time out since no data returns for them. In this way, an IFA can cause severe consumption of both the memory resources of each involved router and the computing resources of target content servers (Afanasyev et al., 2013; Whlisch et al., 2013).

Specifically, as shown in Fig. 1, to guarantee the damage expectation of IFA attackers, the name of each Interest packet is constructed following similar rules: all of the fake Interests should have the same legitimate name prefix (e.g., "/China/Sina/video") yet varying and forged suffixes (e.g., "/attacks1", "/attacks2", etc.). The former guarantees to aggregate as much as malicious traffic, while the latter is to avoid in-network caching hits so that IFA traffic cannot be decreased before they arrive at more victims. For instance, a malicious Interest packet of an IFA with the fake name "/China/Sina/video/attacks1" can be forwarded to the victim *Sina video servers* without being satisfied by intermediate routers, because the forged suffixes guarantee that no content with such a name was cached along the way. To further amplify the damage effect, the fake suffixes of every Interest can also vary randomly to avoid detection (Tourani et al., 2017). In this way, the PIT of each involved router is continually overflown by fake Interests, and meanwhile the victim server unnecessarily wastes time and computation resources to search for the requested fake content against its content index.

**Why an IFA hurts:** In contrast to the convenience of launching such an attack, it is very difficult to detect or mitigate an IFA.

First, ***attacking traffic cannot be accurately identified before it arrives at the victim content servers***, because the attacking traffic is indistinguishable for routers with normal ones. In NDN, there is no difference between legitimate and fake Interest packets of IFA except for the existence of their requested content. This feature of each Interest can only be exactly confirmed by the content servers rather than routers, since only the content servers have all the content and thus can check whether they really exist. Therefore, accurate IFA pre-mitigation on routers is difficult to achieve without the help of content servers in NDN.

Second, ***Interest packets contain no information on the security property of the content name***. The name prefix of each Interest does not contain any security property for its existence verification, which makes accurate detection or traffic filtering very difficult to achieve. Even if an Interest packet is successfully identified as fake in the content servers, this fake name is useless for further IFA mitigation, because content names in IFAs are varied all the time during an attack, and every identified fake name may never be used again to avoid mitigation.

Finally, ***attackers cannot be easily identified or traced to be punished***, since Interest packets in NDN do not carry any information about the requester's identities (Compagno et al., 2013; Gasti et al., 2013) (while in the traditional Internet, the IP address of every content requester is contained in the packet to claim requester's identification (Feng et al., 2017)), which makes attackers able to easily evade from the IFA detection or tracing.

Although the effectiveness of our previous works on countering IFAs (Wang et al., 2014a; 2014b; 2013) has been validated by other parties (Al-Sheikh et al., 2015), we aim here at a further step to achieve a more secure NDN. In this paper, we propose InterestFence, a simple yet efficient IFA countermeasure that involves both *accurate detection at content servers* and *efficient mitigation of ma-*

*licious traffic at intermediate routers, without harming legitimate Interests*. InterestFence filters malicious Interests based on the Hash-based Security Label (HSL) received from content servers. HSL is used to identify whether an Interest packet carries a fake name. Each InterestFence-enabled content server can generate content names following a certain HSL based on some specific algorithms. When an IFA occurs, these content servers determine which name prefix is under attack (denoted $P_i$ as the malicious prefix, meaning Interests with $P_i$ as their name prefix are likely to be fake ones from an IFA), and transmit the $P_i$ and *corresponding HSL algorithm parameters* to the involved routers through an encrypted alarm message. These routers thus are capable of detecting whether an Interest with a specific $P_i$ is fake or not according to the corresponding HSL, and then take corresponding actions, i.e., to drop or forward the Interest packet to the next hop.

The main contributions of this paper can be summarized as follows.

1. The fundamental reasons why an IFA is significantly difficult to detect or mitigate are clearly presented, as well as a comprehensive taxonomy for current IFA countermeasures from the aspects of detection and mitigation.
2. The design detail of InterestFence is given, which enables routers to accurately filter fake Interest packets and directly clean attacking traffic by verifying the HSL of each Interest packet with the infected prefix $P_i$. Owing to its accurate cleaning capability, not only can intermediate routers along the attacking path be protected from an IFA, but the victim content servers can as well. The fake Interest packets are unable to pass the HSL verification in any InterestFence-enabled routers. In this way, both the computation resources of content servers and the memory resources for PIT in each involved router are protected from potential damage caused by an IFA.
3. Extensive experiments on InterestFence were conducted that demonstrates its significant performance and lightweight overhead in accurate IFA detection as well as mitigation. Given the comparative results with state-of-the-art IFA countermeasures, the proposed InterestFence method may be the best one for filtering IFA traffic without harming legitimate requests and content servers.

The rest of the paper is organized as follows. Section 2 provides an overviews of state-of-the-art IFA countermeasures with comprehensive analysis. InterestFence, including its architecture and detailed algorithms, is presented in Section 3. The performance of InterestFence is evaluated in Section 4, and Section 5 concludes this paper.

## 2. Related work

In this section, the brief security threats to NDN are first presented, and then the study of IFA countermeasures is given from two aspects: *detection* and *mitigation*. The former aims at detecting the existence of an IFA, while the latter aims at degrading its damage on critical network resources.

1) **Typical security concerns in NDN**: There are mainly six categories of security vulnerabilities in NDN: IFA, Content Poisoning Attack (CPA), content pollution, secure forwarding, application security, and other subcategories not belonging to any of the preceding ones (Tourani et al., 2017). Recently, a consensus has been reached in the NDN community that IFAs and CPAs have become the two most notorious security threats, the feasibilities of which have been practically proved in real NDN deployments (Mannes and Maziero, 2019; Nguyen et al., 2018). Thus, here we give a comparative analysis for these two significant threats, and those interested in other types of threats can consult the detailed NDN security survey in Tourani et al. (2017).

An IFA aims to overload routers' stateful forwarding plane, while the objective of a CPA is to fill invalid content into routers' caches. The name prefix of the injected content is valid as a normal Interest, but the payload or signature is invalid. In this case, a normal content request from a user may be responded by this invalid cached content from the network, causing bad user experiences as well as unexpected bandwidth consumption. Based on the comprehensive analysis in Tourani et al. (2017) and Nguyen et al. (2018), a single metric (e.g., Interest unsatisfaction ratio) may be enough to be the attack evidence of an IFA. However, it is more difficult to detect a CPA. The most recent design of the NDN security monitoring plane even takes 18 metrics from the PIT, CS, and face component in each router as the input for a correlation engine to identify the abnormal behaviors that indicate a CPA (Nguyen et al., 2018). In this paper, we focus on how to counter an IFA.

2) **IFA detection**: Most of the current methods detect IFAs based on statistics of the router state, such as Interest satisfaction ratio (ISR) (e.g., Afanasyev et al., 2013; Nguyen et al., 2015a; Nguyen et al., 2015b; Salah et al., 2015, Nguyen et al., 2018), number of expired PIT entries in a router (e.g., Wang et al., 2014b), combination of the PIT size and expired PIT entries (e.g., Compagno et al., 2013, Wang et al., 2013), combination of ISR and user reputation (e.g., Umeda et al., 2015), or other factors (e.g., using the information entropy of the Interest names cached in the PIT as one of the IFA detection indicators (Hou et al., 2019; Xin et al., 2016; Zhi et al., 2018a; 2018b)).

In addition, instead of investigating IFAs in a pure NDN architecture, Nguyen et al. (2019) considered a more realistic scenario in which NDN and current IP-based network architecture coexist, and they try to detect an IFA using hypothesis testing theory and evaluate its performance through a real experimental deployment.

However, all of the above detection methods lack accurate IFA identification mechanisms, in that routers do not know exactly whether the name of an Interest is fake or not. For example, most of the existing IFA detection mechanisms may become ineffective when a more sophisticated IFA is launched, which floods Interests with a gradually higher rate that is very low at its beginning to avoid detection (Zhao et al., 2018). Considering that content servers are the final owner of the requested content, they can finally identify an IFA by checking the existence of the requested content. Therefore, InterestFence allocates the detection task to content servers instead of routers during the initialization phase, to guarantee detection accuracy. Then, an HSL for the attacked prefix is sent to corresponding routers to proactively detect the IFA before it reaches victims. Although the works in Dai et al. (2013), Compagno et al. (2015), Liu et al. (2018) and Zhang et al. (2019) also recommend detection of an IFA with the help of content servers, yet they have no such mechanisms as the InterestFence to inform routers of accurate IFA detection operations.

3) **IFA mitigation**: Related solutions can be generally categorized into two groups: the rate-limit mechanism and Interest-PIT decoupling.

The rate-limit mechanism mitigates an IFA by rate-limiting malicious Interests with fake name prefixes (e.g., Wang et al., 2014b, Xin et al., 2016, Umeda et al., 2015) or from interfaces whose satisfaction rate is low (e.g., Dai et al., 2013, Zhang and Li, 2019), or a combination of them (e.g., Afanasyev et al., 2013, Compagno et al., 2015), so that the attacking traffic that can pass routers to reach the final content servers will be reduced. However, in this way, even though the rate of malicious Interest requestss is limited, the rate of legitimate consumers requesting for objects with the same name prefixes is limited as well, which will significantly degrade the quality of the experience for those users.

Interest-PIT decoupling degrades the impact of an IFA on routers by decoupling a huge amount of malicious states related

to fake Interests from the PIT, and thus cuts down the number of ways denial of service can be achieved by exhausting the PIT of NDN routers. Among current Interest-PIT decoupling mechanisms, our previous work in Wang et al. (2013) proposes DPE, which decouples malicious Interest requests from the PIT by appending an interface list to the Interest name, instead of inserting the Interest into the PIT. Similarly, the work in Alston and Refaei (2016) introduces an in-packet cryptographic mechanism called route token that is embedded with incoming interfaces. With route token, there is no need to use the PIT to record pending Interest, as route token can be used to find the incoming interface of a request. In addition, the work in Ghali et al. (2015) also suggests decoupling Interest state from the PIT within each NDN router. Although Interest-PIT decoupling mechanisms successfully prevent an IFA from consuming the PIT resources of intermediate routers, they fail to prevent that malicious traffic from reaching the content servers. As a result, content servers receive all of the malicious traffic.

4) **Summary**: We argue that all of the limitations of currently proposed solutions are rooted in the inaccurate IFA detection at intermediate routers. For Interest packets with the same name prefix, routers cannot distinguish exactly which are from IFA attackers and which come from legitimate consumers, and thus they cannot directly drop all requests but instead can only rate-limit them or decouple them from the PIT as DPE and PIT-less suggested.

In contrast, InterestFence regards IFA detection as the responsibility of content providers, which know exactly the existence of the requested content and can notify the involved routers of the HSL information along the attacking path. With the help of related HSL-verifying operations, intermediate routers can accurately identify and filter all of the fake Interests.

## 3. InterestFence

This section provides the detailed design of InterestFence. First, we introduce the attacking model that InterestFence is designed for, and then describe the system architecture as well as the high-level workflow of InterestFence. Then, we describe how each key component works.

### 3.1. Attacking model

InterestFence is mainly designed for the most promising IFA that aims at causing denial of service for data with certain name prefix(es) (Tourani et al., 2017). For instance, as described in Section 1, the fake Interest packets in an IFA have the same legitimate name prefix but varying and forged suffixes. In this type of IFA, the attacking traffic is easy to aggregate to achieve significant damage on both content servers and NDN routers, as all the fake Interests with the same name prefix will be routed to the victim content server(s) providing data with this prefix (Compagno et al., 2015) via the highly overlapped Internet path. In this case, the number of labels in the malicious list (m-list) module maintained by routers (see Fig. 2) is unlikely to be that large, because the attacker tends to aggregate attacking traffic by using fewer name prefixes.

In fact, except for protecting content servers, InterestFence can also protect routers. Whenever a fake Interest is identified by a content server, the notification will be sent to involved routers. Hence, the first-hop router along the attacking path will filter out attacking traffic, protecting the upstream routers from attacks. Thus, those routers can also be protected.

### 3.2. System overview

Fig. 2 illustrates the high-level architecture of InterestFence, which consists of three key functional entities: InterestFence-enabled router, InterestFence-enabled content server, and the communication channel between them.

First, for each content server, InterestFence adds a *Malicious Prefix Detection* component together with an *HSL Generation* component to generate self-prove content names. Every content name generated by an InterestFence-enabled content server contains a certain HSL as its suffix. An HSL is generated by the *HSL Generation* component, based on some hash algorithms with a secret token for security concerns (e.g., IFA countermeasures). The *Malicious Prefix Detection* component detects IFA attacks by monitoring the requesting statistics of every name prefix periodically. Whenever a malicious prefix $P$ is identified, an alarm message is sent to notify involved downstream routers, to enable HSL validation for all of the Interest packets of content name with the prefix $P$.

Second, from the perspective of each router, InterestFence introduces a m-list module for recording name prefixes under IFA attack, known as $P$, together with their TTLs (e.g., $/P_i$ with $TTL_i$), as well as the corresponding validation tokens (e.g., $\{H_i, K_i\}$) conveyed back by the alarm messages from certain content servers into its *HSL Verification Component*. Moreover, the *TTL* is refreshed whenever a fake Interest is identified by the HSL verification in the router. As shown in Fig. 2, within each InterestFence-enabled router, the actions(e.g., *Forward* or *Drop*) taken on each *suspicious* incoming Interest packet depends on the security property of the incoming Interest (e.g., *fake* or *real*), which can be deduced by the HSL verification results (e.g., *Mismatch* or *Match*).

Finally, between content servers and routers involved in IFA traffic-travelling path, InterestFence needs a "communication channel" for transmitting alarm messages for IFA notification between content servers and routers. The alarm message is a special type of Data packet used for IFA countering purposes, which carries secure information (e.g., cryptographic information containing $\{P_i, H_i, K_i\}$ in Fig. 2) used to verify the content names with the prefix $P_i$, to help make decisions on forwarding or discarding this Interest packet. In its current design, InterestFence takes advantage of the NACK packets (Compagno et al., 2015) to piggy-back alarm messages to the involved routers.

The communication channel is in fact used to indicate the path along which the alarm messages travel from content servers to involved routers. Thus, it is not a real channel like VPN, but a virtual one where routers and content servers communicate with alarm messages. This virtual channel is consistent with the inherit feature of data interaction in NDN, where the Interest/Data exchange naturally forms a virtual channel between consumers and content servers. Let $\{C^*\}$ denote the set of all of the communication channels between servers and routers. For each channel $C_i^*$ $(i \in \mathbb{N})$ between its involved routers $\{R_i^*\}$ and servers $\{S_i^*\}$ along the IFA traffic path, where $C_i^* \in \{C^*\}$, let $\{Msg_i^*\}$ denote all the alarm messages sent along $C_i^*$ to trigger IFA detection and mitigation. Then, we obtain

$$C_i^* \triangleq \{\{R_i^*\}, \{Msg_i^*\}, \{S_i^*\}\} \tag{1}$$

The basic workflow of InterestFence can be described at a high-level as follows.

1) **HSL computation** for every content name in content servers. To announce the existence of some data in NDN, InterestFence-enabled content servers generate legal content names following HSL generating algorithms known only by the content provider, and then sign and publish these names to the public via well-known NDN routing protocols (e.g., NLSR (Hoque et al., 2013; Wang et al., 2018)). The HSL generating algorithms (detailed in

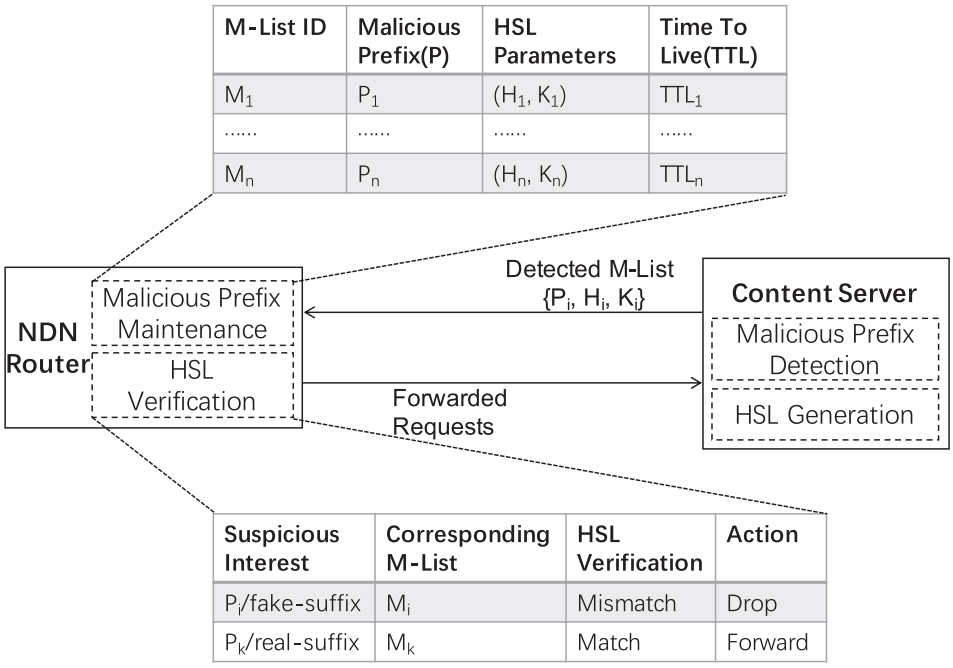



**Fig. 2.** System overview of InterestFence.

Section 3.4) ensure that the algorithm cannot be reversely inferred so that attackers cannot fake legal names.

2) **Identification of malicious name prefixes** when an abnormal number of Interests requesting non-existent content emerges. An InterestFence-enabled content server can easily detect whether it is under IFA attack itself by checking whether request objects exist with the help of its *Malicious Prefix Detection Component*. Afterwards, the content server periodically update the $P_i$ at designated intervals.

3) **Secure HSL transmission** from content servers to involved routers that locate along the attacking path. After certain name prefixes are identified as the $P_i$ under IFA attack, InterestFence-enabled content servers convey corresponding HSL validating algorithms and secret tokens (e.g., $\{P_i, H_i, K_i\}$) back to involved routers along the path in an encrypted manner (detailed in Section 3.5). In InterestFence, non-existing Interests can be labeled malicious only if the number of Interests with the attacked prefix becomes abnormal. In addition, only the first fake Interest packet with the attacked content prefix can trigger an exchange between the targeted content server and involved routers.

4) **IFA traffic filtering** based on the HSL verification component in involved routers. By comparing every Interest name against the m-list via HSL verification, routers know whether a request with $P_i$ is exactly fake or not, and then make a decision to forward or drop the request accordingly. For instance, with reference to Fig. 2, if the verification result of an Interest packet with the content name $/P_i/suffix$ is *HSL Mismatch*, the identification for this Interest is *fake*, and then the suggested operation on it should be *Drop*. As a result, subsequent fake Interests other than the first one with the attacked prefix will be filtered directly by the HSL component implemented in the first-hop router along the same path, and there is no need to trigger any more exchange between servers and routers.

### 3.3. Identification of IFA in servers

The detection of an IFA on a name prefix is conducted by the content server under attack, since only the server itself knows the exact existence of the requested data. Whenever an Interest packet arrives, the content server checks the name of this Interest against all the content items within its memory: if no hit occurs, this Interest is treated as fake, and its name is recorded as $m$.

At the time a fake Interest is identified, the IFA detection period is triggered within this content server. During every monitoring period $t_{decay}$, the server records all of the fake Interests.

At the end of each monitoring period, the malicious name prefix is figured out based on simple operations: Given NDN names are hierarchically structured in the form of "$/ns\_0/ns\_1/ns\_2/.../ns\_k/id$" and the set of name prefixes that are used in HSL computation as $P_{legal}$. The $P_i$ is computed by extracting components of name prefixes from the $P_{legal}$ (that is, $P_i \subseteq P_{legal}$) following the longest matching rules against the received fake Interest names. The detailed pseudo-code for this computation is shown in Algorithm 1. For example, given three fake names $m_1$ = "$/ns\_0/ns\_1/fake$", $m_2$ = "$/ns\_0/ns\_1/ns\_2/malicious$", and $m_3$ = "$/ns\_0/ns\_1/ns\_2/ns\_3/attack$", and the name prefix "$/ns\_0/ns\_1/$" and "$/ns\_0$" belonging to $P_{legal}$, then the detected malicious name prefix $P_i$ = "$/ns\_0/ns\_1/$".

**Algorithm 1** Malicious Prefix Identification.

```
1:  procedure GenMPrefix(names)                    ▷ All fake names
2:      Map nMap ← O          ▷ Key: root prefix; value: name list
3:      for each name in names do
4:          root ← name.rootPrefix
5:          nMap[root].PushInterest(name)
6:      end for
7:      P_i ← O
8:      for each key in nMap.keys do
9:          n ← nMap[key].size()
10:         [m_1, m_2, ..., m_n] ← nMap[key]
11:         tmp_pref ← m_1 ∩ m_2 ∩ ... ∩ m_n
12:         P_i ← P_i ∪ tmp_pref
13:     end for
14: end procedure
```

$$P_i = m_1 \cap m_2 \cap m_3 = /ns\_0/ns\_1 \in P_{legal}. \quad (2)$$

After every monitoring period, each malicious name prefix $P_i$ is transmitted to involved routers to be recorded in their $m_{list}$, through an encrypted method (e.g., asymmetric cryptography technologies described in Section 3.5). It is noted that the overhead caused by encrypted operations is limited (see Fig. 11, at an order of milliseconds and depending on the hardware), because it only needs one cycle of such operation to finish the HSL transmission before an IFA is finished. In addition, whenever no $P_i$ is found in a monitoring period, the IFA detection is disabled.

Quicker detection may be achieved if the routers can actively help to detect an IFA using their history statistic of former network attacks cached in memory. However, it would bring additional overhead to the caching and computing resources of intermediate routers, which may be a serious burden for each involved router since they are the busiest components in NDN. Furthermore, the detection accuracy of an IFA in routers may not be satisfied, because the fake names in malicious Interests vary all of the time, which significantly decreases the effectiveness of performing IFA detection by using the history of attacks cached in routers. Thus, compared to the risk of resource exhaustion as well as low detection accuracy, detecting an IFA directly in content servers may be a more practical choice for network operators.

### 3.4. HSL generation and verification

HSL is fundamentally a wildcard mechanism to validate whether an Interest packet contains a fake name. An Interest packet is treated as fake if its content name cannot match HSL validation.

HSL shares the similar basic idea with digital signature, i.e., the message signed with a private key can be easily validated with the corresponding public-available public key. Digital signature techniques have been studied for decades, and can be used to confirm the integrity of the message. Given that the adversary does not know the private key of content servers, they cannot easily fake a name that can pass the validation with the public key of the provider.

However, using standard digital signature techniques without hardware support introduces high overhead to involved routers and servers considering the frequent usage of verification operations on content names in NDN routers. Consequently, a simple yet efficient enough method, namely HSL, is designed for IFA detection in this paper.

For generating the HSL, a certain hash algorithm will be executed over the chosen bits from the origin name. Then, the suffix will be treated as a signature and used for verifying whether the name is fake. We describe the detailed methodology as follows.

- For each content name $n \in N$ with a certain name prefix in a content server, as shown in Algorithm 2, a certain hash algorithm $H \in H_{algo}$ that takes two parameters $(n,\ K)$, is selected to generate its HSL *hsl* to append to $n$, where $K$ indicates the byte mask used for hash computing. Thus, we obtain $hsl = H(n, K)$ to construct a new content name $n' = g(n, hsl)$, where $g(x,\ y)$ is used to append $y$ to $x$;
- To provide content service, a content server publishes its content name $n'$ to the public, and content consumers use the content name $n'$ to retrieve the data;
- Whenever a name prefix $P_i$ is detected as malicious, the content server transmits the $P_i$ together with the corresponding $\{H_i,\ K_i\}$ to the involved routers in a secure manner (detailed in Section 3.5); for every name prefix, the secured HSL transmission is only executed once, which causes only limited overhead;
- Whenever a router receives an Interest packet, its name prefix is checked against each $P_i$ in the malicious prefix list, as shown in Algorithm 3: if matching, the HSL computation is performed based on the $H_i(n,\ K_i)$, which generates a verifying HSL $hsl'$ for this Interest packet, and then the $hsl'$ is compared with the *hsl* that is originated contained in this Interest packet; if $hsl' \neq hsl$, this Interest packet is fake and thus dropped. Otherwise it is legitimate and passed through to the next hop.

**Algorithm 2** HSL Generation.

```
1: procedure HSLGENERATE(n, H, K, n')
2:     // n: original name, H: hash algorithm chosen from H_algo
3:     // K: wildcard for choosing bits in n, n': new name
4:     hsl ← H(n, K)
5:     n' ← strcat(n, hsl)
6: end procedure
```

**Algorithm 3** HSL Verification.

```
1: procedure HSLVERIFY(n, m_list)
2:     // n: name in Interest request
3:     // m_list: malicious prefix list in router
4:     if prefix of n matches m_list then
5:         P ← m_list[n.prefix]
6:         H ← P.H                    ▷ Get hash algorithm from memory
7:         K ← P.K                    ▷ Get wildcard from memory
8:         hsl ← H(n, K)
9:         hsl' ← suffix(n, len(hsl))
10:        if hsl = hsl' then
11:            return True            ▷ pass validation
12:        else
13:            return False           ▷ fail validation
14:        end if
15:    else
16:        return True                ▷ pass validation
17:    end if
18: end procedure
```

Noting that attackers know neither the $H_i$ nor the $K_i$, thus in theory, they cannot construct the correct *hsl* or the legitimate content name.

With the proposed InterestFence method, legitimate Interests should also be validated in routers if their name prefixes are used for the attack; thus, routers are the busiest components and could become a bottleneck. Fortunately, it has been demonstrated that hash-based verification operations are extremely lightweight in NDN. For instance, the packet forwarding speed based on hash computation can reach 3 million name insertions, lookups, and removals per second in each NDN router, requiring a small amount of memory (Ghasemi et al., 2018). In fact, hash-based verification is even more lightweight than the IP address lookup performed in every router (Luo et al., 2009). Thus, the resource used to validate those Interests with malicious name prefixes is usually limited rather than expensive (see the evaluation results in Section 4.4). This is why we apply hash-based verification in InterestFence. Moreover, in an IFA the attacking traffic has a much larger volume than legitimate Interests to guarantee its damage effect, and thus the resources used for validating legitimate Interests are limited when compared with those for validating IFA traffic.

### 3.5. Encryption mechanism for communication channel

When IFA attacks are detected by a content server, the server does not need to know all of the potentially involved routers. In InterestFence, the content server only needs to send the alarm message in the same way it sends a requested data back. With a PIT,

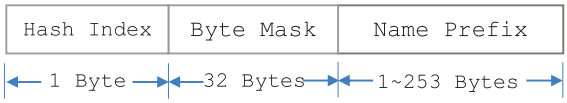



**Fig. 3.** Format of an alarm message.

a router knows where the request comes from and thus is able to convey the alarm message back hop-by-hop. The router is responsible for distinguishing between an alarm message and normal data, and an alarm message can only be transmitted to another router. If the received data are an alarm message, a router will find its neighbouring routers downstream according to the PIT and convey the alarm message back securely. In this way, all involved routers along the path will receive the alarm message.

This solution has two requirements: 1) a router must know whether an interface is connected to another router; 2) a router can talk to its neighbouring router securely. Fortunately, the trust model proposed in Wang et al. (2018) equips each router with a public/private key pair and enables a router to have its neighbouring routers' public keys thus able to talk with each other securely. The proposed mechanism for transferring alarm messages does not require establishing dedicated encrypted channels between content servers and routers potentially involved in IFAs. Instead, alarm messages are exchanged between pairs of routers hop-by-hop. The hop-by-hop exchange mechanism resolves the scalability issue in a large-scale network with relatively long paths.

However, this mechanism does introduce additional overhead because of decrypt/encrypt operations for each exchange between router pairs. To evaluate the proposed mechanism, we investigate the path-length distribution (Teixeira et al., 2003) of a topology derived from CAIDA measurements collected from 16 monitors in 2003, which simulates a network of multiple ASes in the Internet, and conduct experiments to assess the additional overhead of hop-by-hop decrypt/encrypt operations.

According to RFC 1035, the length of a domain name is restricted to 253 octets or less. In NDN, we assume that the name prefix for each content adopts a naming rule similar to today's domain name. Fig. 3 shows the format of an alarm message in InterestFence: (1) the first eight bits indicate the selected hash algorithm among 256 potential candidates, which are pre-defined among all of the routers and content servers; (2) each bit of the *Byte Mask* indicates whether the corresponding byte in the name prefix is selected for the hash algorithm to calculate HSL; and (3) the rest of the message is the *Name Prefix* with a maximum length of 253 bytes.

We ran the RSA-1024 encryption and decryption against the longest alarm message (286 bytes) on a test platform with a Skylake Core-i5 processor(2.7 GHz, 16GB memory). For each pair of decryption and encryption for the name prefix, we conducted the experiment for 100 rounds. The average time consumption of each operation pair was approximately 0.5 ms. Fig. 4 illustrates the distribution of additional time overhead when the alarm message exchange mechanism is applied in the topology derived from CAIDA. It is observed that even when implementing InterestFence in such a realistic Internet topology, over 75% additional time consumption is less than 5 ms in total for an alarm message to be securely transmitted to all involved routers. Furthermore, even for the longest path in this topology, the additional time consumption introduced by InterestFence was only 50 ms.

The measured hopcounts in Teixeira et al. (2003) correspond to a lower bound in the number of paths, while path diversity in such a network is a useful approximation to the real-world Internet. For instance, a more recent study (Ma et al., 2008) shows that the average router-level hopcount of the Internet in China is approximately 16, which shares the same order of magnitude as the result in Teixeira et al. (2003). Consequently, the estimated additional time consumption can be a good approximation for the real-world environment.

The result proves that the overhead of the proposed hop-by-hop alarm message exchange mechanism of InterestFence is insignificant rather than expensive. It brings in very limited time consumption for alarm message exchange, and thus can scale in a large-scale network with relatively long paths, as well as achieve satisfied scalability and feasibility when deployed in a real-world Internet topology.

### 3.6. Update of self-proving content names

It is worth noting that even if the HSL parameters are acquired by attackers, a new $P_{legal}$ for a certain name prefix can also be generated on demand, which in fact consumes limited computing resources and time to finish due to the efficiency of the hash-based verification process (see overhead results in Section 4).

After that, all of the names of this prefix should be re-registered into some name resolution system (Afanasyev et al., 2017), or their reachability information re-announced via popular NDN routing protocols (e.g., NLSR Hoque et al., 2013 Wang et al., 2018), to update their accessibility to anyone in the Internet.

In InterestFence, it is recommended that NDN routing protocols be adopted instead of a name resolution system for updating a content name. The reachability information of all of the content names in each content server can be announced to the public periodically via the NLSR Link State Advertisements (LSAs) messages since NLSR supports dynamic name prefix advertisement and withdrawal (Wang et al., 2018). As a result, whenever a content name is invalid or a new valid self-proving name is generated, its reachability to all of the network entities can be re-distributed within one convergence cycle of the routing protocol.

After the redistribution of the new valid names are accomplished throughout the entire network, all the copies of the content with old names would still be cached in NDN routers, until they are naturally expired. This simple scheme is lightweight as it does not introduce any interactive message for evicting old content, and the Interests from legitimate users can still temporarily retrieve content from the cache of certain NDN routers. Meanwhile, it is secure because in this case, fake Interests constructed based on these old names can be blocked by intermediate routers because the HSL verification is no longer correct in the routers.

## 4. Evaluation

In this section, we provide an in-depth evaluation of InterestFence from three aspects. First, we evaluate the *efficiency* of HSL. Afterwards, we investigate HSL from the perspective of *quality of user experience*. Finally, we compare HSL implementation with potential substitutes owing to a concern with the *trade-off between overhead and security*.

Considering that HSL is the core functional module of InterestFence, we use HSL for short to denote InterestFence throughout this section.

### 4.1. Experiment setup

We developed a simulation platform with senders, routers and servers. Each is configured with parameters like sending rate of senders, capacity and delay of links, capacity of routers, capacity of content servers, and so on. The secure channel for propagating alarm messages from the content server was not included in the experiment due to simulation limitations. Instead, we added an additional delay of 5*ms* for an exchange of alarm messages among routers and servers according to the analysis results in Section 3.5.

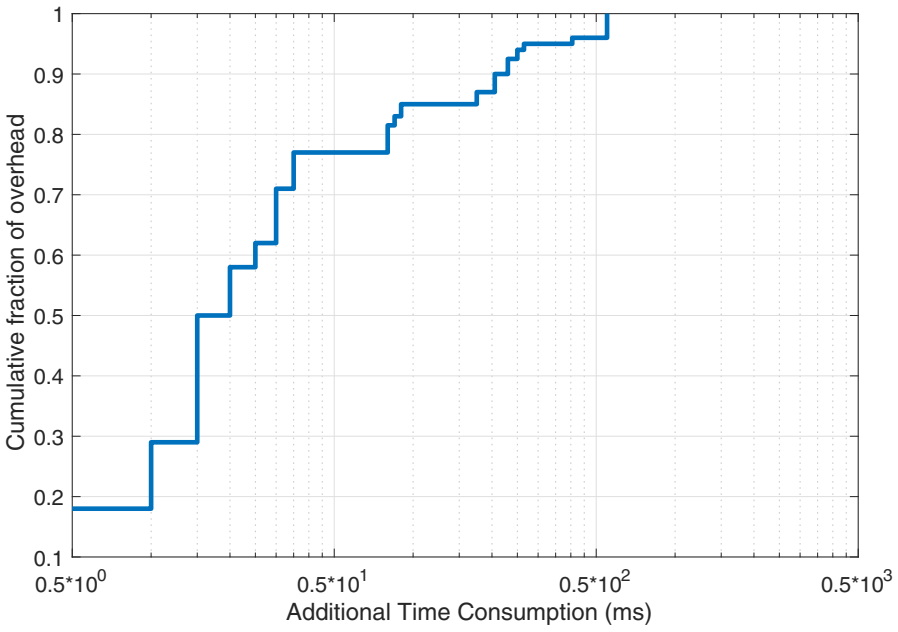



**Fig. 4.** Overhead distribution of virtual channels in topology derived from CAIDA.

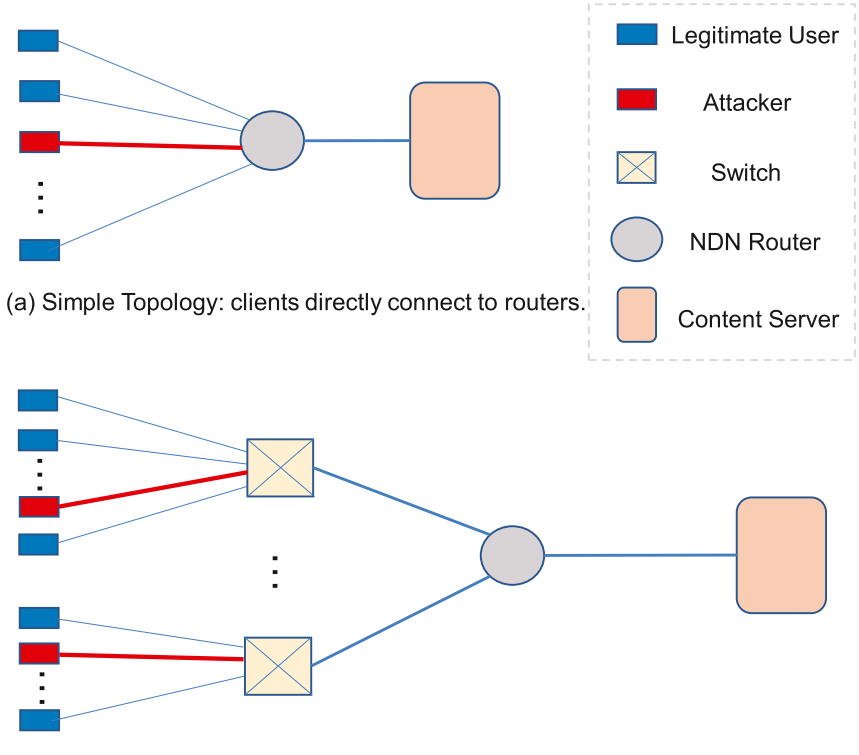



**Fig. 5.** Simulation topology.

**Topology:** We simulated two kinds of many-to-one topology for the experiments, as shown in Fig. 5. For the simple topology, 10 senders including attackers connected to the router directly and then the router connected to the content server. In the complex topology which is more realistic, all of the senders do not connect to a router directly, but they connect to the router through a switch instead. In our experiment, we set up the router connected to 10 switches, each of which had 10 senders attached. The topology used throughout the experiments was the complex one unless otherwise specified. The link capacity from a sender to the router or switch was set to 100 Mbps, while the throughput capacity of the router and server was set to 1 Gbps, which limited the total sending rate of users and attackers. The network RTT for the simple topology was set to 20*ms*, while for the complex topology the RTT was set to 30*ms*.

**Interest sending rate:** In our experiments, senders sent Interest requests following a Poisson Process pattern. The legitimate users sent 100 Interests per second on average, while attackers sent 1000

fake Interests per second. The TTL of an Interest was set to 2000*ms* in the simulation.

**Settings for competitors:** We used the same detection mechanism based on satisfaction ratio of each name prefix at an interface for both rate-limit and PIT-decouple mechanisms. For both rate-limit-based and PIT-decouple-based solutions, the detection parameter $T_r$ was set to 0.8, i.e., a name prefix at an interface with an unsatisfied ratio larger than 80% will be regarded as under attack. A simple but reasonable rate-limit algorithm was implemented in the simulations: whenever an IFA attack is detected, the router will forcibly decrease the corresponding interface's sending rate by half through random dropping.

**Note on $T_r$calculation:** The period $T$ for calculating $T_r$ is derived from RTT between the router and the server. Larger $T$ brings a more accurate satisfaction ratio calculation, while in the meantime it will burden the router when an attack occurs, in that more malicious Interests will be inserted into the PIT before the satisfaction ratio indicates that an attack is taking place. In our experiment, $T$ was set equal to one RTT. In a normal scenario, when no IFA exists, the unsatisfied ratio for a name prefix is ignorable because all the legitimate Interests would be satisfied by returned data within one RTT; that is, if we get a snapshot in each RTT monitoring window, the unsatisfied ratio for a name prefix should be consistently very small, since there is no expired PIT entry in that case. However, if an IFA occurs in the monitoring window, as the fake Interests are continuously recorded in the PIT, there will be an abnormal number of expired PIT entries existing after the TTL of each PIT entry times out. With a current setting of $T_r = 0.8$, the calculation period of each detection mechanism was set as $T = 1RTT$. These simulation settings represent that only serious IFAs causing a large unsatisfied ratio of 80% must be mitigated in our simulations. A smaller value of $T_r$ or larger value of $T$ will bring in a smaller detection time for all of these detection mechanisms, which should be adaptively adjusted by routers or servers in realistic scenarios, according to their resource usage.

### 4.2. Efficiency

In this experiment, the goal was to compare HSL with both rate-limit- and PIT-decouple-based solutions in terms of efficiency. We chose the *percentage of malicious Interests reaching content server (PMR)* as the metric for evaluating efficiency. A higher percentage of malicious Interests reaching content servers results in more severe computation resource consumption, indicating poorer efficiency of an IFA mitigation mechanism.

During the experiments, we first investigateed the PMR with a constant percent of attackers, and then varied the percentage of attackers from 10% to 70% in the network to collect more comprehensive statistics at the server side.

Fig. 6 illustrates the dynamics of the PMR value during and after an IFA with 50% attackers existing in network. The attack started from the tenth second and lasted till the end. Prior to start of the attack, all the PMRs were set to zero. For all three methods, the value built up rapidly as soon as the IFA occurred and Interests started to time out. However, PMR value of three solutions differed significantly in the duration of the attack: the PMR of DPE remained 100% until the attack stopped, while rate-limit PMR decreased slightly within several RTTs. HSL was quite different, as its PMR directly decreased to zero after several RTTs.

This result shows that HSL can almost filter out all of the malicious traffic as soon as the server detects the attack and notifies the involved router, while DPE only decouples malicious traffic from the PIT and rate-limit-based solution just reduces the rate of affected interfaces at routers.

It is interesting that the DPE method leads systematically to 100% of malicious traffic passing through, because its design philosophy is to decouple malicious states from the PIT to reduce the router's burden, whereas all of the malicious Interests are not recorded in the PIT any more, yet are carried in the name of each malicious Interest itself for data response. Thus, the PIT consumption in each involved router can be kept almost the same as normal conditions without attacks, but all of the malicious Interests travel through routers to content servers. Thus, DPE is useful for countering the IFA targeting routers rather than content servers.

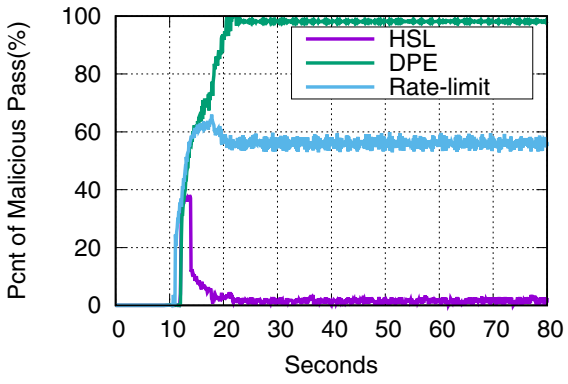
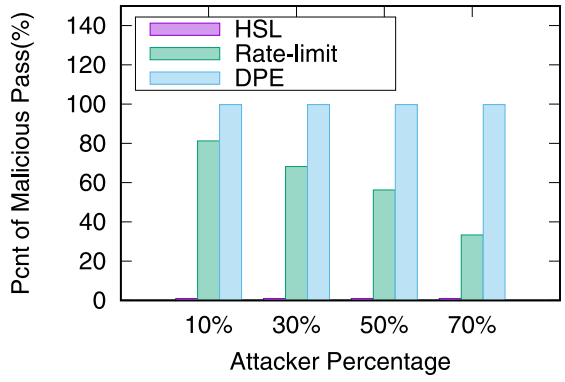
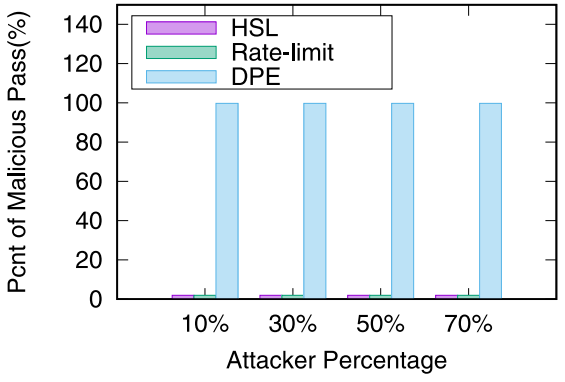



**Fig. 6.** Dynamics of malicious Interest reaching content server statistics (with 50% attackers in network).




(a) Results in complex topology




(b) Results in simple topology

**Fig. 7.** Comparison of malicious Interest pass percent of different mechanisms under various attack burdens.

As shown in Fig. 7, we changed the percentage of attackers and compared the PMR value at steady state during the attack.

Fig. 7(a) shows the results in the complex topology. As the result shows, HSL always kept PMR to 0, while DPE always kept it to 100%. For the rate-limit solution, as the percentage of attack-

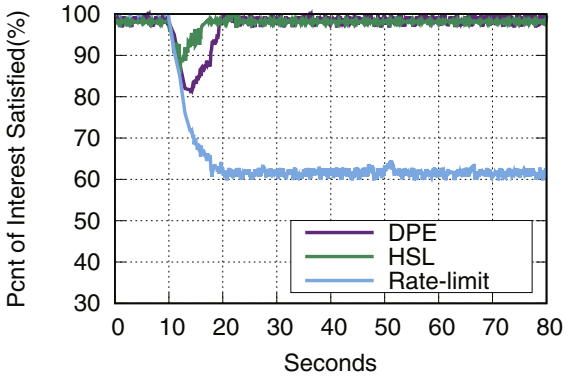



**Fig. 8.** Dynamics of the percentage of satisfied Interests (with 30% attackers in network).

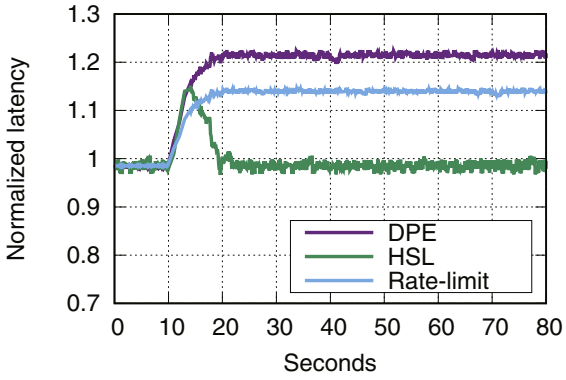



**Fig. 9.** Dynamics of latency with different mechanism under attack (with 30% attackers in network).

ers in the network increases, the PMR value decreases. This is because as the percentage of attackers increases, the unsatisfied ratio is more likely to be larger than $T_r$ and rate limiting occurs more often. In this case, the ratio of dropped Interests increases, which decreases the PMR. In Fig. 7(b), the result of HSL and DPE in the simple topology is identical to that in the complex topology. However, for the rate-limiting mechanism, as the attackers are connected to the router directly, the router will continuously decrease the rate of affected interfaces until the rate reaches zero. In this scenario, the attacker is completely isolated and the rate-limiting mechanism achieves a result as good as that achieved by HSL. Nevertheless, the simple topology scenario is very rare in that client hosts usually do not connect to an NDN router directly.

These two experiments demonstrate that HSL is quite efficient in filtering malicious traffic and protecting routers and content servers from IFAs.

### 4.3. Quality of experience

In terms of quality of experience, we chose the *percentage of satisfied Interests (PSI)* for legitimate users as the metric. This metric quantifies the quality of service experienced by legitimate users when the network is under attack. For two different methods *A* and *B*, if legitimate users of the network equipped with method *A* achieve a higher percentage of user-satisfied Interests while the network is under attack than that of the network equipped with method *B*, then one can conclude that method *A* is more effective than method *B* at mitigating the IFA attack.

During the experiment, we collected statistics at the sender side to calculate the PSI of legitimate users.

Fig. 8 shows the dynamics of the PSI value during and after an IFA with 30% attackers in a network under protection of each IFA countermeasure. Prior to the start of the attack, all of the PSIs are 100%. The PSI value decreases rapidly as soon as an IFA occurs and Interests start to time out. However, similar to PMR, the PSI value of the three solutions differs significantly during the attack: the PSIs of HSL and DPE both slightly drop, i.e., by 10% and 20%, respectively, and recover nearly 100% within several RTTs. However, the PSI of the rate-limiting solution decreases more severely and cannot recover until the attack stops.

In terms of latency, as Fig. 9 illustrates, the latency of HSL slightly spikes at the beginning of an attack and recovers to normal values quickly. However for DPE, the latency becomes high when an attack occurs and cannot return to the normal level. This is because in DPE all attack Interests reach the content server and consume computation resources of the server, which results in higher latency. The latency of the rate-limiting solution is similar to that of DPE, but is a little better. This is because some malicious Interests are rate-limited by the routers so that the content server receives fewer attacks and thus fewer resources are consumed by malicious traffic.

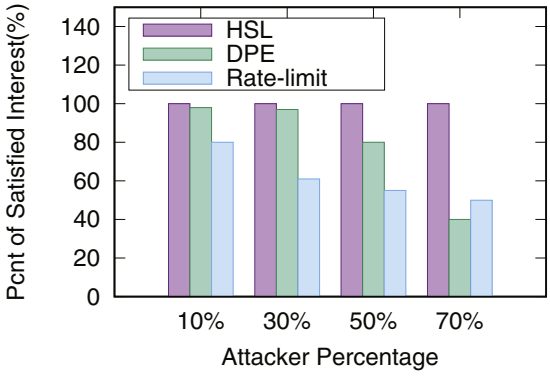



**Fig. 10.** Comparison of Interest satisfied percent of different mechanisms under various attack burdens.

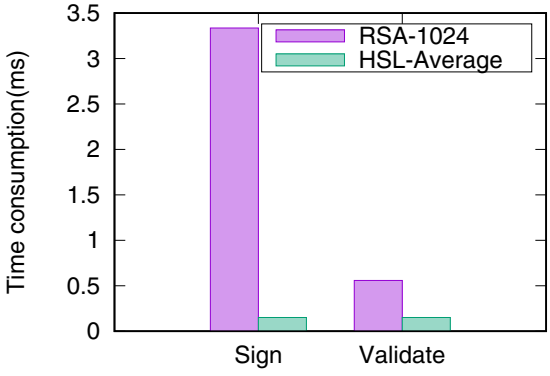



**Fig. 11.** Overhead of time consumption.

As shown in Fig. 10, we further changed the percentage of attackers and comapared the PSI value at steady state during an attack. As shown in Fig. 10, HSL always keeps the PSI at nearly 100%. The rate-limiting solution obtains a smaller PSI value as the percentage of attackers in the network increases. An interesting result is that, while DPE keeps the PSI at nearly 100% when the attack burden is not very high, when the percentage of attackers becomes extremely high DPE starts to perform even worse than rate-limiting solutions. This is because a high load of malicious Interests reaching the victim server begins to totally exhaust its computation resources.

These experiments prove that HSL can ensure the quality of user experience during IFA attacks significantly better than existing methods. DPE performs poorer than HSL in that all malicious Interests are forwarded to content servers and the server uses more computation resources to fight against the attack. DPE performs better than the rate-limiting solution because intermediate routers are freed from malicious Interests in DPE.

### 4.4. Overhead

Considering the high frequent usage of validation in network scenarios, a tradeoff is required between the overhead and security level. HSL shares a similar idea with digital signature mechanism, while it is much simpler in terms of computational complexity. In this experiment, we compared HSL with a typical digital signature technique, asymmetric RSA signature (RSA-1024), from the aspects of both overhead and security.

This experiment required no network setup, but compared the computational resource consumption. We chose *time per-signature* and *time per-validation* as the metrics to evaluate the overhead. Higher time per-signature or per-validation all denote higher overhead. In terms of security, we used *false positive* and *false negative* to measure security.

The dataset used in this experiment comprised 100,000 URLs we crawled from Sina, one of the top content providers in China. We classified these URLs based on their sub-domains and transformed them into the form of an Interest name in NDN.

During the experiment, we set up a single thread with adequate memory resources. We read into a thousand URLs into the memory at one time and signed the URLs one by one separately with RSA and HSL. We then calculated the average per-signature time of RSA and HSL. Similarly, we obtained the average per-validation time of RSA and HSL, correspondingly.

The result is shown in Fig. 11. As can be seen, RSA takes as much as 30 times longer in signing and 5 times more time in validation than HSL, indicating that RSA consumes much more computational resources.

We then compared RSA and HSL in terms of security. For the 100,000 URLs, we randomly generated names for each name prefix as the fake name for attackers. We wanted to calculate the percentage of fake names that can pass HSL validation.

In the experiment, one can see that HSL has a possibility of 1% to mistakenly treat a randomly generated fake name as a legal one. In other words, HSL trades a false positive rate of 1% to achieve several tens of performance increase. We do not claim that HSL provides a security guarantee like that provided by RSA and other complex signature mechanisms, but we state that it is worthwhile to make a sacrifice for performance by using a comparatively simple hashing algorithm. If a system has a very strict requirement for security concern, RSA and similar complex but more secure algorithms should be used instead of a simple hashing algorithm.

## 5. Conclusions

In this paper, we presented InterestFence, which is an efficient IFA mitigation framework that can accurately identify fake Interests and efficiently filter attacking traffic at intermediate routers. It has two key contributions: *(i)* a fast and accurate HSL generating component at content servers, and *(ii)* a lightweight and accurate name verification component at routers. We performed extensive evaluations for InterestFence using simulations with comprehensive analysis of the results, which indicate that it is efficient in both IFA detection and mitigation. InterestFence can filter out 100% of the malicious traffic at intermediate routers, and achieves the same level of legitimate Interest satisfaction ratio as without IFAs, consuming very low overhead.

In planned future work, the blockchain technology will be embedded into the InterestFence, to enable an incentive for routers to offload workload and counter an IFA with higher efficiency.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2016YFB1000102), National Natural Science Foundation of China (No. 61702439, 61972222, 61602030), Shandong Provincial Natural Science Foundation of China (No. ZR2017BF018).

## References

Afanasyev, A., Jiang, X., Yu, Y., Tan, J., Xia, Y., Mankin, A., Zhang, L., 2017. NDNS: a DNS-like name service for NDN. In: Proceedings of the 26th International Conference on Computer Communications and Networks (ICCCN), pp. 1–9. Vancouver, BC, Canada

Afanasyev, A., Mahadevan, P., Moiseenko, I., Uzun, E., Zhang, L., 2013. Interest flooding attack and countermeasures in named data networking. In: Proceedings of IFIP Networking, pp. 1–9. Brooklyn, NY, USA

Afanasyev, A., Moiseenko, I., Zhang, L., 2012. NDNSIM: NDN Simulator for NS-3. Technical Report NDN-0005. NDN.

Al-Sheikh, S., Whlisch, M., Schmidt, T.C., 2015. Revisiting countermeasures against NDN interest flooding. In: Proceedings of the 2nd ACM Conference on Information-Centric Networking (ACM-ICN), pp. 195–196. San Francisco, CA, USA

Alston, A., Refaei, T., 2016. Neutralizing interest flooding attacks in named data networks using cryptographic route tokens. In: Proceedings of IEEE 15th International Symposium on Network Computing and Applications (NCA), pp. 85–88. Cambridge, MA, USA

Carofiglio, G., Gallo, M., Muscariello, L., Perino, D., 2011. Modeling data transfer in content-centric networking. In: Proceedings of 23rd International Teletraffic Congress (ITC), pp. 111–118. San Francisco, CA, USA

Carofiglio, G., Gallo, M., Muscariello, L., Perino, D., 2015. Pending interest table sizing in named data networking. In: Proceedings of the 2nd ACM Conference on Information-Centric Networking (ACM-ICN), pp. 49–58. San Francisco, California, USA

Carofiglio, G., Morabito, G., Muscariello, L., Solis, I., Varvello, M., 2013. From content delivery today to information centric networking. Comput. Netw. 57 (16), 3116–3127.

Compagno, A., Conti, M., Gasti, P., Tsudik, G., 2013. Poseidon: mitigating interest flooding DDOS attacks in named data networking. In: Proceedings of IEEE 38th Conference on Local Computer Networks (LCN), pp. 630–638. Sydney, NSW, Australia

Compagno, A., Conti, M., Ghali, C., Tsudik, G., 2015. To nack or not to nack? Negative acknowledgments in information-centric networking. In: Proceedings of the 24th International Conference on Computer Communication and Networks (ICCCN), pp. 1–10. Las Vegas, NV, USA

Dai, H., Wang, Y., Fan, J., Liu, B., 2013. Mitigate DDOS attacks in NDN by interest traceback. In: Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 381–386. Turin, Italy

Dong, J., Wang, K., Lyu, Y., Jiao, L., Yin, H., 2018. Interestfence: countering interest flooding attacks by using hash-based security labels. In: Proceedings of International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), pp. 527–537. Guangzhou, China

Feng, B., Zhang, H., Zhou, H., Yu, S., 2017. Locator/identifier split networking: a promising future internet architecture. IEEE Commun. Surv. Tut. 19 (4), 2927–2948.

FIA-NP: Collaborative Research: Named Data Networking Next Phase (NDN-NP). https://www.nsf.gov/awardsearch/showAward?AWD_ID=1345286.

Gasti, P., Tsudik, G., Uzun, E., Zhang, L., 2013. DOS and DDOS in named data networking. In: Proceedings of 22nd International Conference on Computer Communication and Networks (ICCCN), pp. 1–7. Nassau, Bahamas

Ghali, C., Tsudik, G., Uzun, E., Wood, C. A., 2015. Living in a PIT-less World: A Case Against Stateful Forwarding in Content-Centric Networking. arXiv:1512.07755.

Ghasemi, C., Yousef, H., Shin, K.G., Zhang, B., 2018. A fast and memory-efficient trie structure for name-based packet forwarding. In: Proceeding of 26th IEEE International Conference on Network Protocols (ICNP 2018), pp. 1–11. Cambridge, UK

Hahm, O., Baccelli, E., Schmidt, T.C., Whlisch, M., Adjih, C., Massouli, L., 2017. Low-power internet of things with NDN & cooperative caching. In: Proceedings of the 4th ACM Conference on Information-Centric Networking (ACM-ICN), pp. 98–108. Berlin, Germany

Hoque, A.K.M.M., Amin, S.O., Alyyan, A., Zhang, B., Zhang, L., Wang, L., 2013. NLSR: named-data link state routing protocol. In: Proceedings of the 3rd ACM SIGCOMM Workshop on Information-Centric Networking (ICN 2013), pp. 15–20. Hong Kong, China. doi: 10.1145/2491224.2491231.

Hou, R., Han, M., Chen, J., Hu, W., Tan, X., Luo, J., Ma, M., 2019. Theil-based countermeasure against interest flooding attacks for named data networks. IEEE Netw. 33 (3), 116–121. doi:10.1109/MNET.2019.1800350.

Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M., Briggs, N., Braynard, R., 2012. Networking named content. Commun. ACM 55 (1), 117–124.

Kurose, J., 2014. Information-centric networking: the evolution from circuits to packets to content. Comput. Netw. 66, 112–120.

Liu, G., Quan, W., Cheng, N., Wang, K., Zhang, H., 2018. Accuracy or delay? A game in detecting interest flooding attacks. Internet Technol. Lett. 1 (2), 1–6. doi:10.1002/itl2.31.

Liu, X., Yang, X., Xia, Y., 2010. Netfence: preventing internet denial of service from inside out. In: Proceedings of ACM SIGCOMM, pp. 255–266. New Delhi, India

Luo, H., Qin, Y., Zhang, H., 2009. A DHT-based identifier-to-locator mapping approach for a scalable internet. IEEE Trans. Parallel Distrib.Syst. 20 (12), 1790–1802. doi:10.1109/TPDS.2009.30.

Ma, J., Xi, M., Lin, Y., Li, Y., 2008. Chinese internet routerlevel hop count measurement and analysis. Appl. Res. Comput. 25 (27), 2112–2114.

Mangili, M., Martignon, F., Capone, A., 2016. Performance analysis of content-centric and content-delivery networks with evolving object popularity. Comput. Netw. 94, 80–88.

Mannes, E., Maziero, C., 2019. Naming content on the network layer: a security analysis of the information-centric network model. ACM Comput. Surv. 52 (3), 1–28.

Mastorakis, S., Afanasyev, A., Zhang, L., 2017. On the evolution of NDNSIM: an open–source simulator for NDN experimentation. ACM Comput. Commun. Rev. 47 (3).

Ngai, E., Ohlman, B., Tsudik, G., Uzun, E., Whlisch, M., Wood, C.A., 2017. Can we make a cake and eat it too? A discussion of ICN security and privacy. ACM SIGCOMM Comput. Commun. Rev. 47, 49–54.

Nguyen, T., Cogranne, R., Doyen, G., 2015. An optimal statistical test for robust detection against interest flooding attacks in CCN. In: Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 252–260. Ottawa, ON, Canada

Nguyen, T., Mai, H.-L., Cogranne, R., Doyen, G., Mallouli, W., Nguyen, L., Aoun, M.E., de Oca, E.M., Festor, O., 2019. Reliable detection of interest flooding attack in real deployment of named data networking. IEEE Trans. Inf. Forensics Secur. 14 (9), 2470–2485. doi:10.1109/TIFS.2019.2899247.

Nguyen, T., Mai, H.-L., Doyen, G., Cogranne, R., Mallouli, W., de Oca, E.M., Festor, O., 2018. A security monitoring plane for named data networking deployment. IEEE Commun. Mag. 56 (11), 88–94. doi:10.1109/MCOM.2018.1701135.

Nguyen, T.N., Cogranne, R., Doyen, G., Retraint, F., 2015. Detection of interest flooding attacks in named data networking using hypothesis testing. In: Proceedings of IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6. Rome, Italy

Posch, D., Rainer, B., Hellwagner, H., 2017. SAF: stochastic adaptive forwarding in named data networking. IEEE/ACM Trans. Netw. 25 (2), 1089–1102.

Quan, W., Xu, C., Guan, J., Zhang, H., Grieco, L.A., 2014. Social cooperation for information-centric multimedia streaming in highway vanets. In: Proceeding of IEEE WoWMoM Workshop, pp. 1–6. Sydney, NSW, Australia doi: 10.1109/WoWMoM.2014.6918992.

Salah, H., Wulfheide, J., Strufe, T., 2015. Lightweight coordinated defence against interest flooding attacks in NDN. In: Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 103–104. Hong Kong, China

Su, Z., Hui, Y., Yang, Q., 2017. The next generation vehicular networks: acontent-centric framework. IEEE Wirel. Commun. 24 (1), 60–66.

Teixeira, R., Marzullo, K., Savage, S., Voelker, G.M., 2003. Characterizing and measuring path diversity of internet topologies. In: Proceedings of the International Conference on Measurements and Modeling of Computer Systems (ACM SIGMETRICS), pp. 304–305. San Diego, CA, USA. 10.1145/781027.781069.

Tourani, R., Misra, S., Mick, T., Panwar, G., 2017. Security, privacy, and access control in information-centric networking: a survey. IEEE Commun. Surv. Tut. PP (99), 1–36.

Umeda, S., Kamimoto, T., Ohata, Y., Shigeno, H., 2015. Interest flow control method based on user reputation and content name prefixes in named data networking. In: Proceedings of IEEE Trustcom/BigDataSE/ISPA, pp. 710–717. Helsinki, Finland

Wang, K., Chen, J., Zhou, H., Qin, Y., Zhang, H., 2014. Modeling denial-of-service against pending interest table in named data networking. Int. J. Commun. Syst. 23 (12), 4355–4368.

Wang, K., Zhou, H., Luo, H., Guan, J., Qin, Y., Zhang, H., 2014. Detecting and mitigating interest flooding attacks in content-centric network. Secur. Commun. Netw. 7 (4), 685–699.

Wang, K., Zhou, H., Qin, Y., Chen, J., Zhang, H., 2013. Decoupling malicious interests from pending interest table to mitigate interest flooding attacks. In: Proceedings of IEEE Globecom Workshops (GC Wkshps), pp. 963–968. Atlanta, GA, USA

Wang, L., Lehman, V., Hoque, A.K.M.M., Zhang, B., Yu, Y., Zhang, L., 2018. A secure link state routing protocol for NDN. IEEE Access 6, 10470–10482. doi:10.1109/ACCESS.2017.2789330.

Whlisch, M., Schmidt, T.C., Vahlenkamp, M., 2013. Backscatter from the data plane - threats to stability and security in information-centric network infrastructure. Comput. Netw. 57 (16), 3192–3206.

Xin, Y., Li, Y., Wang, W., Li, W., Chen, X., 2016. A novel interest flooding attacks detection and countermeasure scheme in NDN. In: Proceedings of IEEE Globecom, pp. 1–7. Washington, DC, USA

Xylomenos, G., Ververidis, C.N., Siris, V.A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K.V., Polyzos, G.C., 2014. A survey of information-centric networking research. IEEE Commun. Surv. Tut. 16 (2), 1024–1049.

Zargar, S.T., Joshi, J., Tipper, D., 2013. A survey of defense mechanisms against distributed denial of service (DDOS) flooding attacks. IEEE Commun. Surv. Tut. 15 (4), 2046–2069.

Zhang, H., Quan, W., chieh Chao, H., Qiao, C., 2016. Smart identifier network: a collaborative architecture for the future internet. IEEE Netw. 30 (3), 46–51. doi:10.1109/MNET.2016.7474343.

Zhang, X., Li, R., 2019. A charging/rewarding mechanism-based interest flooding attack mitigation strategy in NDN. In: Proceedings of 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 402–407. Arlington, VA, USA

Zhang, Z., Lung, C.-H., Lambadaris, I., St-Hilaire, M., 2017. When 5g meets ICN: an ICN-based caching approach for mobile video in 5g networks. Comput. Commun.

Zhang, Z., Vasavada, V., K, S. K. R., Osterweil, E., Zhang, L., 2019. Expect more from the networking: DDOS mitigation by fitt in named data networking. 1–15, arXiv:1902.09033.

Zhao, L., Cheng, G., Hu, X., Wu, H., Gong, J., Yang, W., Fan, C., 2018. An insightful experimental study of a sophisticated interest flooding attack in NDN. In: Proceedings of 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN 2018), pp. 121–127. Shenzhen, China

Zhi, T., Liu, Y., Yan, Z., 2018. An entropy-SVM based interest flooding attack detection method in ICN. In: Proceedings of 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pp. 1–5. Chicago, IL, USA. 10.1109/VTCFall.2018.8690809.

Zhi, T., Luo, H., Liu, Y., 2018. A gini impurity-based interest flooding attack defence mechanism in NDN. IEEE Commun. Lett. 22 (3), 538–541.