An Overview of Content Poisoning in NDN: Attacks, Countermeasures, and Direction

Hyeonseung Im and Dohyung Kim*
Department of Computer Science and Engineering, Kangwon National University, Chuncheon, South Korea
[hsim@kangwon.ac.kr, d.kim@kangwon.ac.kr]
*Corresponding author: Dohyung Kim

Received February 12, 2020; revised April 2, 2020; accepted May 24, 2020; published July 31, 2020

Abstract

With a huge demand for replicated content on the Internet, a new networking paradigm called information-centric networking (ICN) has been introduced for efficient content dissemination. In ICN, named content is distributed over the network cache and it is accessed by name instead of a location identifier. These aspects allow users to retrieve content from any of the nodes having replicas, and consequently 1) network resources are more efficiently utilized by avoiding redundant transmission and 2) more scalable services are provided by distributing server loads. However, in-network caching in ICN brings about a new type of security issues, called content poisoning attacks, where fabricated content is located in the network cache and interferes with the normal behavior of the system. In this paper, we look into the problems of content poisoning in ICN and discuss security architectures against them. In particular, we reconsider the state-of-the-art schemes from the perspective of feasibility, and propose a practical security architecture.

Keywords: Information-centric networking, Network cache, Content poisoning attacks, Content verification, Verification overhead

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2018R1C1B4A01022931, NRF-2019R1F1A1063272) and 2019 Research Grant from Kangwon National University

http://doi.org/10.3837/tiis.2020.07.010 **ISSN : 1976-7277**

1. Introduction

Today's Internet has been designed based on TCP/IP protocols, which was first introduced in the late 1960s to provide host-to-host data communication. For decades, the Internet has successfully evolved according to users' changing demands. However, with the explosively increasing numbers of mobile devices and massive video services, it faces the fundamental limitations arising from host-to-host communication. Since all requests are served by only a few designated servers, increased server loads may incur a scalability issue. In addition, network resources are wasted by repeated transmission of the same content over the same link.

To resolve this problem, an information-centric networking (ICN) paradigm has emerged as an alternative. In ICN, named content is distributed over the network and it is accessed by name, not by the location identifier. Consequently, any nodes having replicas can serve user requests, which enables that content is fetched in more flexible and efficient ways [1-8]. Under this networking model, however, fabricated/poisoned content\footnote[1] can also be easily disseminated, and it poses a security threat to end users as well as networking systems. This type of security attack is called content poisoning.

A content poisoning attack is launched by using either compromised routers or collaboration between end-hosts. Once fabricated content is put into a network node, it has the potential to serve user requests after matching content names. The fabricated content may exhaust storage and computation resources of routers while being delivered to the user. To prevent poisoned content from spreading across the network, a digital signature is embedded in every piece of content and it is verified either at routers or end-hosts. However, since the verification process incurs a large overhead, it is not feasible that routers verify all content at the line rate. The existing literature [9-19] mainly addressed how to reduce/avoid such verification overhead at routers.

According to the tactical objective, the existing studies can be categorized into three groups: lightweight validation, selective verification, and detouring. Schemes, which fall within the category of light-weight verification, aim to present a new validation system that can replace expensive operations for signature verification. In selective verification schemes, routers verify only a subset of content that is either suspicious or popular. The rationale behind these schemes is to minimize meaningless/unnecessary verification. Detouring approaches focus on making alternative paths to the valid content, naturally avoiding routers having poisoned content.

In this paper, we carefully discuss merits and demerits of all these state-of-the-art schemes in terms of feasibility, and conclude that (1) any single approach has limitations in its effectiveness and/or implementation and (2) a practical security architecture can be deployed when the existing schemes are effectively combined. To present the direction of the research, we summarize the common views obtained from the existing literature. Based on those common views, an example of combined architectures is sketched and discussed. We believe that the simplicity and effectiveness of the proposed design contribute to implementing a practical security architecture against content poisoning attacks in ICN.

The rest of this paper is organized as follows. Section 2 introduces a Named-Data Networking (NDN), which is a representative architecture of ICN, and the content poisoning

[1] Fabricate/poisoned content refers to the content that has a valid name which could be matched with the corresponding request, but its payload or signature is faked.

attack in NDN. In Section 3, previous studies in the literature are categorized according to the tactical objective, and the pros and cons of each category are discussed in terms of feasibility and practicality. Based on the lessons of this survey, we carefully select modules and sketch the practical security architecuture in Section 4. Then, the effectiveness of the proposed sketch is discussed and evaluated theoretically. Finally, Section 5 concludes the paper.

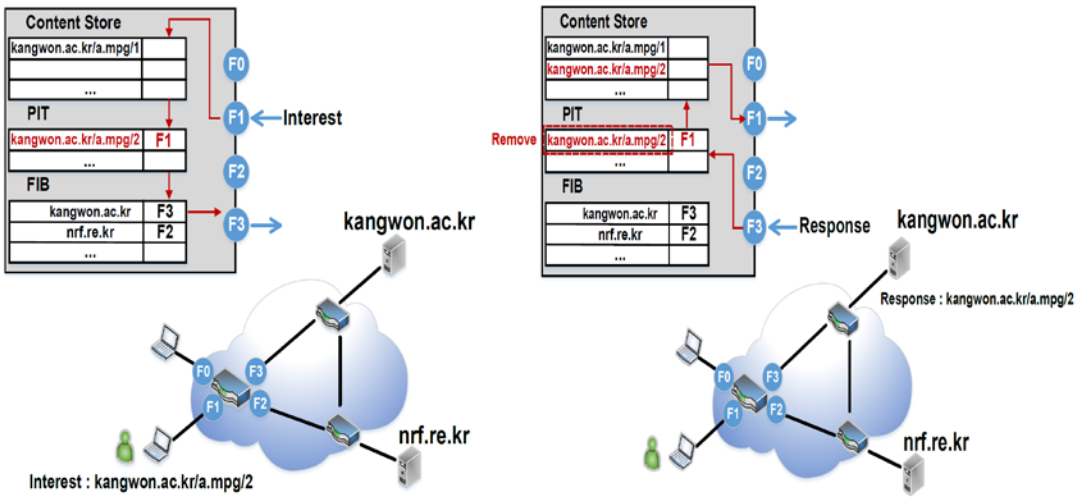

Fig. 1. Routing and caching in NDN routers

2. Preliminaries

2.1 Named-Data Networking Architecture

As massive content retrieval becomes prevalent in the Internet, ICN has been designed with the objectives of fast and efficient content distribution and access [20]. Among several ICN candidates [1-6], a Named-Data Networking (NDN) architecture has attracted great attention from the academia and industry. In NDN, a receiver initiates communication by issuing a request for named content, called an INTEREST. An INTEREST is routed by the content name toward the nodes having either the target content or its replicas. When an INTEREST arrives at each router, the router looks up the target content in its local cache (called CONTENT STORE) by the content name specified in the INTEREST. If the target content is found, the INTEREST stops propagating further and the target content is served by the CONTENT STORE. Otherwise, the INTEREST is stored in Pending Interest Table (PIT) with the incoming face. For example, in **Fig. 1(a)**, F1 is stored with the content name in PIT. Then, the corresponding NDN data packet (called RESPONSE) would be delivered to the user along the same reverse path. For example, in **Fig. 1(b)**, the RESPONSE (kangwon.ac.kr/a.mpg/2) is forwarded via F1 by consuming its corresponding PIT entry. When an arrives at a router, if an entry for the same content already exists in PIT, the incoming face is simply added to the PIT entry and the INTEREST is discarded. Otherwise, a new PIT entry is created and the INTEREST is routed to the next hop based on Forwarding Information Base (FIB). When the RESPONSE is delivered to the user, the content is cached in the CONTENT STORE of intermediate routers.

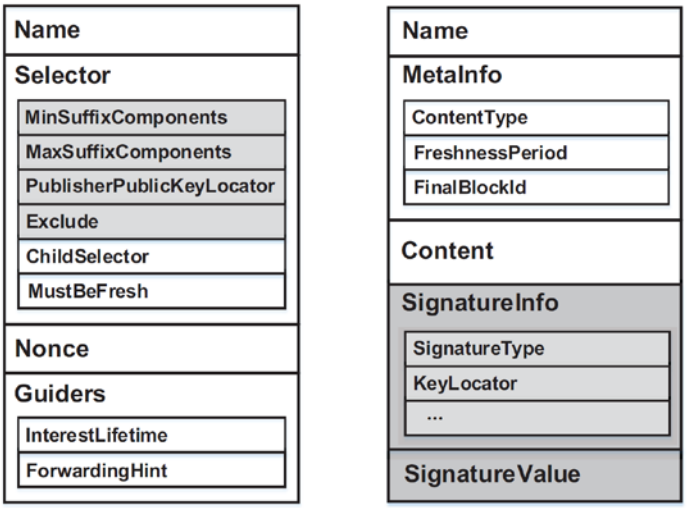

Fig. 2. NDN packet format and security relevant fields

2.2 Poisoned Content in NDN

In IP networks, a security attack mainly concerns about how to disrupt communication channels. For example, in the attacks of IP hijacking and spoofing, attackers masquerade as legitimate nodes by manipulating IP addresses, steal connections, or fool naive users into accepting bogus data. To resolve this problem, two communication end-hosts must secure the channel using methods such as Secure Sockets Layer (SSL). In NDN, however, content could be served not only by designated servers, but also any nodes in the network. Even routers become potential content providers by installing the CONTENT STORE. Due to this characteristic, even the definition of a secure channel is ambiguous in NDN.

Instead of communication channels, NDN secures content itself. Every piece of content includes a digital signature that can be verified either by routers or end-hosts as shown in **Fig. 2(b)**. More specifically, content provider generates a key pair (public key and private key) using a key-generation algorithm, and content is digitally signed by the content provider's private key. Here, digital signature schemes such as RSA and ECDSA are used. The signature is calculated over the entire content (Name, MetaInfo, Content, and SignatureInfo) and specified in the field of *SignatureValue* (Fig. 2(b)). This signature can be verified using the corresponding public key, which is generally fetched via the network as a piece of another NDN data (*SignatureInfo* contains the information used to verify the content, e.g, signature algorithm and information needed to obtain the parent certificates). Hence, the public key must also be authenticated before the signature verification. A possible approach to build trust of this public key is to use a Public Key Infrastructure (PKI) [21]. This approach prevents poisoned content from being inserted into the CONTENT STORE. Naturally, it gets rid of the threat that poisoned content contaminates CONTENT STORE of routers while propagating toward the content requester.

However, it has been reported in the literature [14] that an unrealistic number of computing resources is required for wire-speed verification over line-rates in the order of hundreds Gbps. Given that NDN routers work extremely busy with routing, caching, and cache-lookup, the verification overhead can be a serious challenge.

2.3 Implementation of Content Poisoning Attacks and Its Effects

To implement content poisoning attacks, poisoned content may be inserted into the CONTENT STORE by using compromised routers or malicious end-hosts [16]. One of the strong benefits of NDN is that any nodes having the copies of content can serve user requests, but it also implies that attackers' traffic may be injected into the network with ease. For example, in the content discovery phase, an INTEREST may reach an attacker's server and poisoned content is served as a response. If an attacker's nodes are used to issue the INTEREST as well, poisoned content would get into the network more easily. Since verification is not mandatory for routers due to a large overhead, poisoned content is placed in the CONTENT STORE of intermediate routers during its transport (see **Fig. 3**).

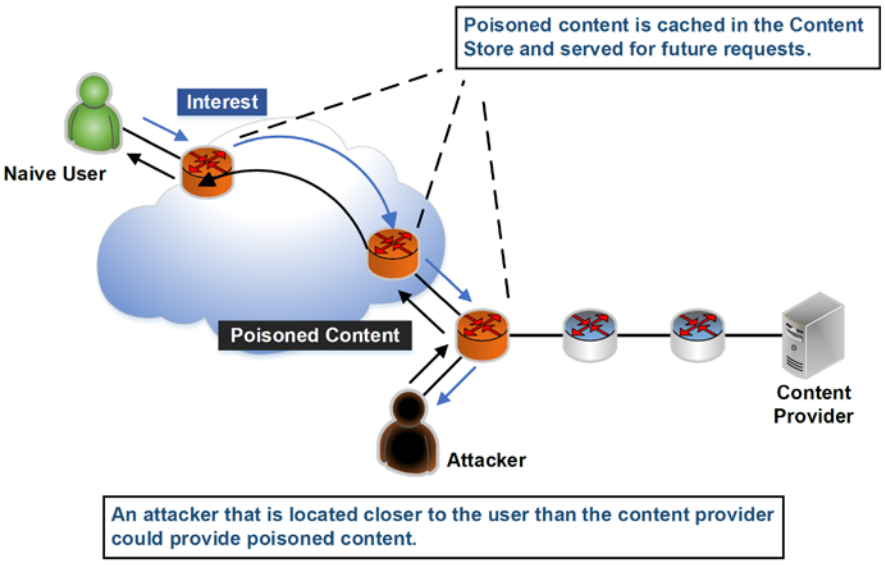

Fig. 3. Content Poisoning Attacks

Once poisoned content appears in the CONTENT STORE, it may be accessed by future requests based on simple name matching and easily distributed over the network. Poisoned content wastes limited space on the CONTENT STORE. In addition, it disturbs users to reach valid content with a single attempt. If a user fails to validate the received poisoned content, he or she reissues an INTEREST that has the hash value of the received poisoned content in the field of *Exclude*. Routers would serve the INTEREST with the cached content having different hash values or forward the INTEREST to the next hop. This process incurs an additional access delay and hash checking overhead at routers.

3. Countermeasures for Content Poisoning Attacks in NDN

Various schemes have been introduced in the literature to tackle the issue of verification overhead. We categorize them into three groups: lightweight validation, selective verification, and detouring.

3.1 Lightweight Validation

The research direction of lightweight validation is to design a new verification system that can replace costly computations that are used to verify digitally signed content.

3.1.1 Hashing Content

The use of self-certifying name in NDN has been discussed in [9]. When a RESPONSE is created, the hash value that is computed over the packet (content, name, signature) is added to the content name as a suffix. If users know this hash value beforehand and specify it in the request message (INTEREST), routers can check the integrity of content by matching the hash values in the INTEREST and RESPONSE. The authors in [10] have proposed a similar approach using Interest-Key Binding (IKB). IKB addresses that "INTEREST must reflect the public key of the producer." To use the IKB rule in practice, the digest of the content producer's public key is enclosed in the INTEREST (in the field of PublisherPublicKeyLocator (PPKL) (see **Fig. 2(a)**). When content is delivered to the user, the RESPONSE has a verification key itself in the field of KeyLocator, instead of any reference to the key or a certificate having the key. Hence, on receiving a RESPONSE, routers hash the public key in the filed of KeyLocator and check whether it matches with the value of PPKL in the pending INTEREST. If they are not matched with each other, the content is discarded.

According to the official NDN documentation [22], SHA256 hash is used to provide several types of signature. In general, the choice of a hash function between Modification Detecton Code (MDC) and Message Authentication Code (MAC) depends on the application. If MAC should be generalized for better security, the secret key can be shared using the trust anchor as explained in [16]. The trust anchor could be a form of a global Key Name Server (similarly to today's DNSSEC) or organization-specific trust anchor (similar to the SDSI [23]). Since comparing hash values requires much less computation than verifying a signature [24], validation overhead is effectively reduced in these schemes. However, a key challenge of these schemes is how users obtain hash values or public keys of what they request in advance. This problem has been considered a chicken and egg situation since hash values must also be securely provided [12]. In [10], a few solutions have been discussed upon the assumption of a *well-defined trust management architecture*. For example, NDN applications may have pre-installed root public keys, which are used to request lower-level certificates for each content. A global Key Name Service (which is similar to the Domain Name Service in the IP network) or search engines like Google may be alternatives to retrieve public key certificates for the public key names. The success of these solutions depends on the implementation of a well-defined trust management system between users and content providers.

3.1.2 User Feedback

Instead of performing signature verification, routers analyze feedback from users to distinguish valid and poisoned content [11, 12] . In NDN, all users would verity the signature of the received content. When verification fails, users re-request content by issuing a new INTEREST. That INTEREST includes the hash value of the poisoned content in the field of *Exclusion*. When a router receives an INTEREST having the hash value of poisoned content, it lowers the rank of the corresponding content in its CONTENT STORE. As a result, valid content is likely to be ranked higher than poisoned content, and it would be served for future requests. This approach effectively avoids huge verification overhead. However, the main

drawback in this approach is that the ranking mechanism could be overwhelmed by false reports from malicious users. For example, false reports can have hash values for valid content, so router can delete valid content [11] or lower the ranking of valid content than those of poisoned content [12].

3.2 Selective Verification

The objective of selective verification is to reduce the number of verification while keeping the integrity of cached content.

3.2.1 Probabilistic Caching

In [13, 14], content is verified and inserted into the CONTENT STORE with a certain probability. As the probability becomes lower, a less amount of content is picked up for being verified and cached. Consequently, more popular content is likely to be inserted in the CONTENT STORE, while a smaller amount of overhead occurs. Despite such advantages, however, a low insertion probability incurs a recency problem in the presence of dynamic content since outdated content could stay longer in the CONTENT STORE. Given that the optimal insertion probability varies depending on the traffic pattern as shown in [15], it is difficult to control the insertion probability under dynamic network conditions.

3.2.2 Verification on a Cache-hit

In [15], every content is inserted into the CONTENT STORE without being verified. Later, when content is served from the CONTENT STORE, its signature is verified. This approach is based on the observation that a large amount of content is not accessed again and simply expelled from the CONTENT STORE. Since such content has little effect on the network, its verification is considered meaningless. With further optimization techniques, this approach could significantly reduce the verification overhead. However, it is vulnerable to the attack where a massive number of cache hits are manipulated by malicious users to overwhelm the router's authentication system. In [16], the solution to this problem was proposed using the correlation between the amount of cache-hit content and the number of cache-hit events. Since a few pieces of popular content is repeatedly accessed in general, the number of cache-hit events is much larger than the number of pieces of cache-hit content. However, if routers are under attack, a large amount of different content may be used to generate cache-hit events. (Here it is noted that verification is performed only on the first cache hit of the content.) Then, the ratio between verification numbers and cache-hit events would abnormally increase. Through the proposed solution in [16], routers could detect and prevent attacks by simply counting the number of verification and cache-hit events.

3.2.3 Verification for the Reported Content

Feedback from neighbors (either routers or users) has also been used for selective verification. In [14, 17, 18], when poisoned content is detected, a special warning INTEREST is generated and transmitted to all neighbor nodes. On receiving the warning INTEREST, routers selectively perform verification on that reported content, and continue to propagate the warning INTEREST if the verification fails. Similar to the other feedback-based schemes,

however, this approach may not function properly due to false reporting from malicious users. For example, attackers may generate massive warning INTERESTs to ensure that the router (e.g., edge router in [18]) runs out of computational resources while performing signature verification.

3.3 Detouring

Detouring approaches mainly discuss how to establish alternative paths to the valid content while isolating malicious routers.

3.3.1 Reputation-based Forwarding

In [19], a reputation-based forwarding scheme has been designed to localize malicious routers. In this scheme, each router maintains a reputation per its neighboring node (per face). The reputation linearly increases on receiving valid content. When a report of poisoned content is delivered from users, the reputation of the neighboring node that conveyed poisoned content decreases exponentially. If the reputation of a neighboring node is less than a certain threshold, the router do not forward INTEREST toward the neighboring node until its reputation would become restored. Since routers rely on user reports to validate content, this scheme is also vulnerable to false reports from malicious users. Upon receiving false reports of valid content, specifically, routers may reduce the reputation of innocent routers and exclude them from the network. False reports themselves could be used to waste computational resources on routers.

3.3.2 Forwarding over Hop-by-Hop Trust Management

Under hop-by-hop trust management, routers exchange user reports for poisoned content [17,18]. Each router can validate a report independently because the report contains not only poisoned content but also the sender's identification. On receiving a valid report, routers remove poisoned content in the CONTENT STORE and adjust routing paths to avoid contaminated routers. However, creating and relaying poisoned content cannot be differentiated under this scheme. Hence, naive routers that relay poisoned content could be excluded from the network [17] or not fully exploited [18] just because they comply with the routing rule of NDN. Besides, the report is very large (since it includes poisoned content and information to authenticate the sender) and it requires to be validated. Hence, the report itself could be used to implement another type of denial-of-service (DoS) attacks.

Table 1. Comparisons of previous studies

Approach		Pros	Cons	Related Works
Lightweight Verification	Hashing	- Less Computational overhead	- Trust management architecture is required	[9,10,12]
	User feedback	- No verification on the router	- Vulnerable to false feedback	[11,12]
Selective Verification	Prob. Caching	- Likely to verify a limited number of popular content	- Hard to find the optimal prob. (recency problem)	[13,14]
	On cache-hit	- Minimizing unnecessary verification	- No action to adjust the path to the valid content	[15,16]
	When reported	- Verifying only the reported content	- Vulnerable to false reports	[14,17,18]

Detouring	Reputation -based forwarding	- No verification on the router	- Vulnerable to false reports - Naive routers could be excluded	[19]
	Forwarding over trust management	- Resilient to false reports	- Validation overhead for the report - Naive routers could be excluded	[17,18]

4. Direction for Practical Solution

While revisiting the state-of-the-art solutions to mitigate content poisoning attacks, we have seen that any single approach has limitations either in its effectiveness or implementation. **Table 1** summarizes the pros and cons of each approach discussed in Section 3.

However, each scheme obviously introduced a meaningful idea that could be collectively used to build a more practical security architecture against content poisoning attacks. As the first step toward such a combined architecture, we first summarize the common views that have been learned from the existing literature as follows:

- Cryptographic operations to verify signatures are considered too expensive for routers. Using either a digest of the content (hash value) or users' feedback may be feasible alternatives in practice.
- Not every content needs to be verified. If cached content has little impact on the network (it does not serve an INTEREST), its validation is a waste of resources.
- Propagation of user reports (feedback) has a potential risk of another type of DoS.

By taking these lessens into consideration, we propose an example of combined architectures. The overall design of the proposed architecture is described in **Fig. 4**. It is composed of a verification module, a monitoring module, and a controlling module. In the verification module, two different validation methods are applied depending on the type of payload, that is, signature verification for the catalog and hashing-based verification for the content. It also supports delayed verification where content is verified on the cache-hit event. The monitoring module is added to detect whether the router is under attack or not and to identify vulnerable faces that are connected to an attacking node. The controlling module manipulates the forwarding preference in FIB to avoid paths including attacking nodes. In the following subsections, we describe how and when the proposed architecture verifies content. In addition, it is also specified that how attacking nodes are localized without the overhead of message exchanging.

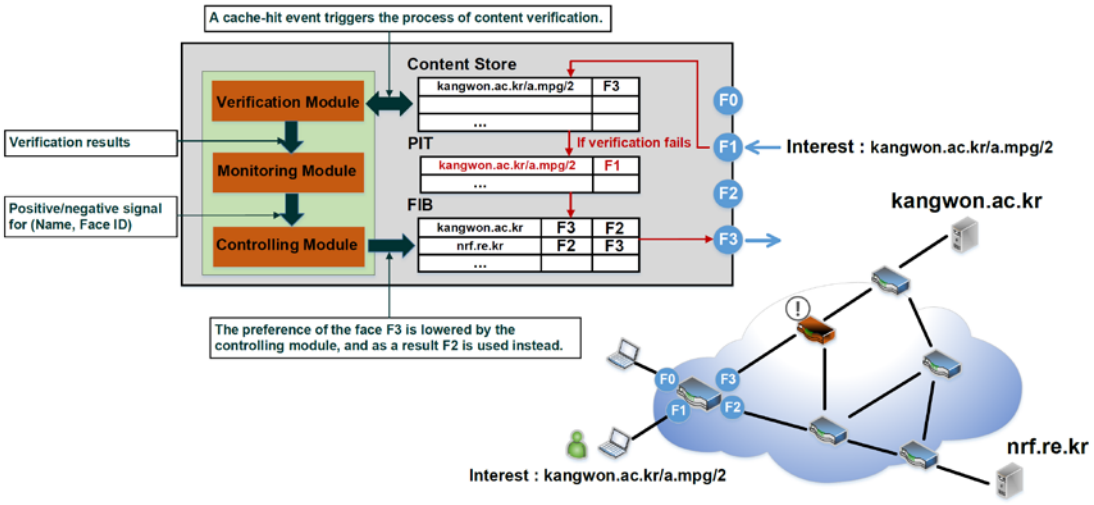

Fig. 4. An example of combined architecture

4.1 Digitally Signed Catalog

Performing a validity check using hash values is a lightweight alternative to expensive signature verification. The challenge of this approach is to establish a trust management system that securely provides hash values of what users request in advance. To avoid such a need, we suggest that a catalog is included as the first piece of content (e.g., ContentName/catalog). The catalog contains hash values for all pieces of content. It is digitally signed by the producer and verified by the signature even at router. After routers obtain a valid catalog, they validate content by hashing every piece of received content and comparing the result with the hash value in the catalog.

To implement this scheme, each RESPONSE has a flag to indicate "This is a catalog". This flag is cached with content and used later when the content is validated by the verification module. Here, the NDN may flexibly change the path to the content, so some routers can receive some pieces of the content without getting a catalog. When any pieces of content need to be verified but the router does not have the catalog, the verification process begins by importing the catalog using the known name ContentName/catalog and verifying it. The catalog remains cached in the CONTENT STORE until the last piece of corresponding content is evicted. Because the catalog is validated by signature verification, this approach works without an additional trust management system required for users to retrieve key digests or hash values in advance. Meanwhile, since most parts of the content are validated by matching hash values, verification overhead may be reduced significantly.

4.2 Delayed Verification

We borrow the delayed verification scheme that has been introduced in [15]. Content is validated when it is accessed by the INTEREST rather than when it is inserted into the CONTENT STORE. Using this strategy, content that is not served from the CONTENT STORE is not validated, thus saving a large number of computational resources.

When content is cache-hit, the verification module performs a signature verification of the content catalog first. Each piece of content is then validated by using the hash value in the

verified catalog. Note that the verification process for each piece of content is performed only once when the first cache hit occurs. After the content has been validated successfully, a flag stating that the content is valid is specified in the CONTENT STORE. The flag subsists while the content stays in the CONTENT STORE. Hence, subsequent INTERESTs access the content without performing a verification process.

4.3 Isolation of Attacking Nodes by Local Monitoring

A user report may be used to detect and isolate attacking nodes, but it creates additional overhead needed to validate the report itself. Because of this overhead, it can be used to implement other types of security attacks that exhaust computing resources on routers. Hence, instead of exchanging messages such as user reports, we employ a local monitoring technique to exclude attacking nodes from the network.

In the proposed approach, the validation result is notified to the monitoring module from the verification module. The monitoring module keeps track of failed verification (the ratio of the failed verification to the overall verification trials). If it exceeds a threshold, the monitoring module determines that the router is under attack and moves into the inspection phase where vulnerable faces are identified. In the inspection phase, the monitoring module counts the number of received poisoned content per face. If the percentage of poisoned content received from a particular face is larger than a threshold, the monitoring module gives a negative signal with the ID of the face to the controlling module. The controlling module then lowers the forwarding preference of that reported face in FIB, which prevents subsequent INTERESTs from being forwarded through the face.

The monitoring module periodically checks the vulnerable face by replicating INTERESTs, but not too often. If the content delivered from the vulnerable face is successfully validated, a positive signal and the ID of that vulnerable face are sent from the monitoring module to the controlling module. The positive signal additively increases the score of the vulnerable face. If the score reaches a threshold value, the forwarding preference of the vulnerable face is restored. Here, it is noted that any single notification of failed verification initializes the score to zero immediately. Since validation results are not propagated over the network, routers would be free from a security concern that might arise from a manipulated feedback report.

4.4 The Effectiveness of the Example Architecture

The performance gain from the digitally signed catalog and delayed verification can be estimated based on the previous work. In [14], hashing-based verification reduced verification overhead by up to one-tenth, compared to signature verification. In [15], delayed verification saved computing resources that were used to validate bypassing cached content [2] that accounted for 90% of the total traffic.

To identify the effectiveness of the local monitoring scheme, we performed simulation by using the ndnSIM simulator (https://ndnsim.net/current/). The simulation topology is shown in **Fig. 5(a)**. User requests were created in the same pattern as the YouTube trace on the UMASS campus from March 11-17, 2008 [25]. The overall number of pieces of content in the trace was 158,974, and it was assumed that all content was the same size. The sum of users' INTEREST generation rates was 400 pieces per second and the size of CONTENT STORE was set to 1000.

[2] Bypassing cached content refers to the content that is evicted from the CONTENT STORE without serving an INTEREST.

The ratio of poisoning content at the compromised router was varied from 0 to 0.5. Simulation was performed for 1000 seconds, and records for the last 500 seconds were analyzed.

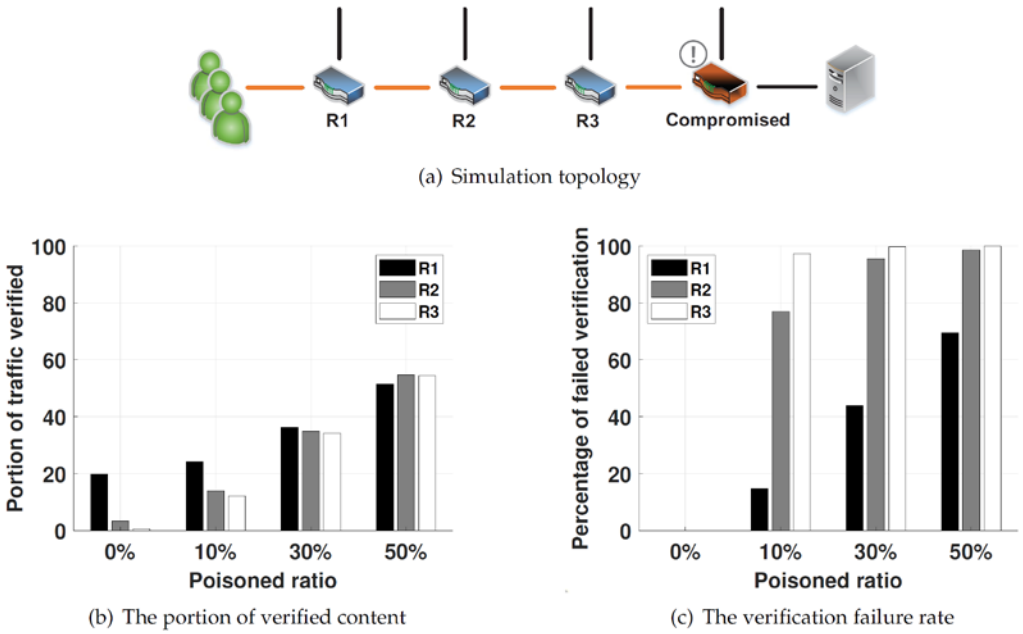

Fig. 5. Simulation results

In the proposed system, a cache hit was occurred either when popular content was repeatedly accessed or poisoned content was requested again by the user who failed to validate the received content. Since more popular content was stored closer to users under LRU cache replacement, more (successful) verification (for the valid popular content) was observed on R1 (edge router) if no content is poisoned (refer to the graphs for 0% in **Fig. 5(b)**). With poisoned content appearing, the verification overhead has increased in proportion to the ratio of poisoned content on all routers, which is shown in **Fig. 5(b)**.

Here, the verification failure rate, defined as

$$F(n) = \frac{V_f(n)}{V_f(n) + V_s(n)}$$

was monitored on each router and the results are described in **Fig. 5(c)** ($V_f(\boldsymbol{n})$ and $V_s(\boldsymbol{n})$ denote the number of verification for poisoned content and valid content on the router $\boldsymbol{n}$, respectively.). The verification failure rate was reduced proportionally to the hop distance to the attacker, resulting in R_1 (edge router) having the smallest value. This is because the edge router typically has a higher cache-hit rate for popular content, which results in a larger value of $V_s(\boldsymbol{n})$.

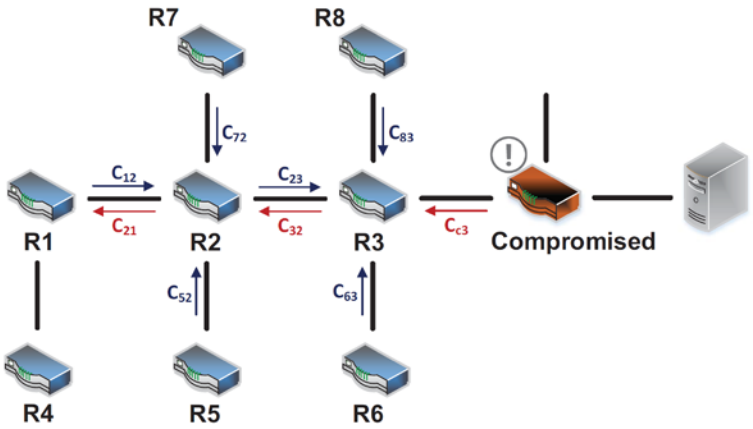

Fig. 6. Part of a typical network topology

In the general topology such as in **Fig. 6**, $F(R_1)$ may be greater than $F(R_3)$. The cache on R_3 can be filled with more popular content and $V_s(R_3)$ becomes larger than $V_s(R_1)$. In that situation, R_1 or R_2 may step into the inspection phase earlier than R_3. However, R_3 would be the first to identify the vulnerable face in the inspection phase since the proportion of poisoned content delivered through the vulnerable face is the largest at R_3. This is obvious because C_{32} (or C_{21}) may additionally contain content delivered from other faces such as C_{63} and C_{83} (or C_{52} and C_{72}) other than C_{c3}. Therefore, $P(C_{c3}) \geq P(C_{32})$ (or $P(C_{21})$), where $P(\boldsymbol{x})$ represents the proportion of poisoned content out of all content $\boldsymbol{x}$.

By using the proposed scheme, R3 can disable the vulnerable face and help subsequent INTERESTs circumvent the attacking node. Then, no more poisoned content would be delivered from the compromised router, and the percentage of failed verification at the other victim routers[3] would also decrease. Consequently, victim routers would not be excluded from the network. Here it is noted the portion of verified traffic was quite close to the poisoning rate at the compromised router. This result implies that the system minimized unnecessary validation attempts and effectively detected poisoned content.

5. Conclusion

In this paper, we revisit the stat-of-the-art schemes which mitigate the impact of content poisoning attacks, and discuss their merits and demerits from the viewpoint of feasibility and cost. From this study, we learn that any single approach has limitations either in its effectiveness or implementation, and a practical security architecture can be built when the existing schemes are effectively combined. As the first step toward the future research direction, we arrange the common views that have been addressed in the existing literature: 1) The hash value can be used to reduce the verification overhead 2) Unnecessary validation can be minimized by considering cache characteristics 3) Verification techniques that rely on other network nodes can be vulnerable to another type of security attacks. Based on these common views, an example of combined architectures against content poisoning attacks is sketched and discussed. Currently, the example architecture is being complemented and refined. In future

[3] Victim routers are the routers that simply relay poisoned content, not creating it. In **Fig. 5(a)**, R_1, R_2, and R_3 are victim routers.

work, the details of the proposed architecture will be introduced and its empirical study using real-world network topologies and request traces will be presented to discuss the scalability issue. We hope that our survey and concept proposal in this paper would be shared by the academia and industry and contribute to building a practical security architecture in ICN.

References

[1] Koponen, Teemu, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica, "A data-oriented (and beyond) network architecture," in *Proc. of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 181-192, 2007. Article (CrossRefLink)

[2] Jacobson, Van, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard, "Networking named content," in *Proc. of the 5th international conference on Emerging networking experiments and technologies*, pp. 1-12, 2009. Article (CrossRefLink)

[3] Tarkoma, Sasu, Mark Ain, and Kari Visala, "The Publish/Subscribe Internet Routing Paradigm (PSIRP): Designing the Future Internet Architecture," *IOS Press*, pp. 102-111, 2009. Article (CrossRefLink)

[4] Dannewitz, Christian, "Netinf: An information-centric design for the future internet," in *Proc. of 3rd GI/ITG KuVS Workshop on The Future Internet*, pp. 1-3, 2009. Article (CrossRefLink)

[5] García, Gerardo, Andrzej Beben, Francisco J. Ramón, Adrián Maeso, Ioannis Psaras, George Pavlou, Ning Wang et al, "COMET: Content mediator architecture for content-aware networks," in *Proc. of 2011 Future Network & Mobile Summit*, pp. 1-8, 2011. Article (CrossRefLink)

[6] Lagutin, Dmitrij, Kari Visala, and Sasu Tarkoma, "Publish/Subscribe for Internet: PSIRP Perspective," *IOS Press*, 75-84, 2010. Article (CrossRefLink)

[7] Vasilakos, Athanasios V., Zhe Li, Gwendal Simon, and Wei You, "Information centric network: Research challenges and opportunities," *Journal of network and computer applications*, 52, 1-10, 2015. Article (CrossRefLink)

[8] White, Greg, and Greg Rutz, "Content delivery with content-centric networking," *CableLabs, Strategy & Innovation*, 1-26, 2016. Article (CrossRefLink)

[9] Jacobson, Van, Jeffrey Burke, Deborah Estrin, Lixia Zhang, Beichuan Zhang, Gene Tsudik, Kimberly Claffy et al., "Named data networking (NDN) project 2012-2013 annual report," *Named Data Networking (NDN)*, 2013. Article (CrossRefLink)

[10] Ghali, Cesar, Gene Tsudik, and Ersin Uzun, "Network-layer trust in named-data networking," *ACM SIGCOMM Computer Communication Review*, 44, no. 5, 12-19, 2014. Article (CrossRefLink)

[11] Cui, Wenjing, Yang Li, Yan Zhang, Chang Liu, and Mengqi Zhan, "An Ant Colony Algorithm Based Content Poisoning Mitigation in Named Data Networking," in *Proc. of 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)*, pp. 176-183, 2019. Article (CrossRefLink)

[12] Ghali, Cesar, Gene Tsudik, and Ersin Uzun, "Needle in a haystack: Mitigating content poisoning in named-data networking," in *Proc. of NDSS Workshop on Security of Emerging Networking Technologies (SENT)*, 2014. Article (CrossRefLink)

[13] Bianchi, Giuseppe, Andrea Detti, Alberto Caponi, and Nicola Blefari Melazzi, "Check before storing: What is the performance price of content integrity verification in LRU caching?," *ACM SIGCOMM Computer Communication Review*, 43, no. 3, 59-67, 2013. Article (CrossRefLink)

[14] Gasti, Paolo, Gene Tsudik, Ersin Uzun, and Lixia Zhang, "DoS and DDoS in named data networking," in *Proc. of 2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pp. 1-7, 2013. Article (CrossRefLink)

[15] Dohyung Kim, Sunwook Nam, Jun Bi, and Ikjun Yeom, "Efficient content verification in named data networking," in *Proc. of the 2nd ACM Conference on Information-Centric Networking*, pp. 109-116, 2015. Article (CrossRefLink)
[16] Dohyung, Kim, Jun Bi, Athanasios V. Vasilakos, and Ikjun Yeom, "Security of cached content in NDN," *IEEE Transactions on Information Forensics and Security*, 12, no. 12, 2933-2944, 2017. Article (CrossRefLink)
[17] DiBenedetto, Stephanie, and Christos Papadopoulos, "Mitigating poisoned content with forwarding strategy," in *Proc. of 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 164-169, 2016. Article (CrossRefLink)
[18] Cui, Wenjing, et al, "Feedback-based content poisoning mitigation in named data networking," in *Proc. of 2018 IEEE Symposium on Computers and Communications (ISCC). IEEE*, 2018. Article (CrossRefLink)
[19] Wu, Danye, Zhiwei Xu, Bo Chen, and Yujun Zhang, "What if routers are malicious? mitigating content poisoning attack in ndn," in *Proc. of 2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 481-488, 2016. Article (CrossRefLink)
[20] Ahlgren, Bengt, Christian Dannewitz, Claudio Imbrenda, Dirk Kutscher, and Borje Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, 50(7), 26-36, 2012. Article (CrossRefLink)
[21] Hamdane, B., Serhrouchni, A., Fadlallah, A., & El Fatmi, S. G., "Named-data security scheme for named data networking," in *Proc. of 2012 Third International Conference on The Network of the Future (NOF)*, pp. 1-6, 2012. Article (CrossRefLink)
[22] "NDN Packet Format Specification 0.3 document," 2020. [Online]. Available: https://named-data.net/doc/NDN-packet-spec/current/
[23] R. L. Rivest and B. Lampson, "SDSI–a simple distributed security infrastructure," *Crypto*, 1996.
[24] Wang, Yi, Zhuyun Qi, Kai Lei, Bin Liu, and Chen Tian, "Preventing" bad" content dispersal in named data networking," in *Proc. of the ACM Turing 50th Celebration Conference-China*, pp. 1-8, 2017. Article (CrossRefLink)
[25] "Youtube traces from the campus network," 2008. [Online]. Available: http://traces.cs.umass.edu/index.php/Network

Hyeonseung Im received the B.S. degree in computer science from Yonsei University, Korea, in 2006 and the Ph.D. degree in computer science and engineering from Pohang University of Science and Technology (POSTECH), Korea, in 2012. From 2012 to 2015, he was a Postdoctoral Researcher with the Laboratory for Computer Science at Université Paris-Sud and with the Tyrex team, Inria, France. Since 2015, he has been an Assistant Professor with the Department of Computer Science and Engineering, Kangwon National University, Korea. His research interests include programming languages, logic in computer science, big data analysis and management, machine learning, precision medicine, and network security.

Dohyung Kim received the BS degree in information and computer engineering from Ajou University, Suwon, South Korea, in February 2004, and the PhD degree in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in August 2014. He was a postdoc researcher and research professor with the Department of Computer Engineering, Sungkyunkwan University from 2014 to 2017. In 2018, he was an assistant professor with the Department of Software and Computer Engineering, Ajou University. Currently, he is an assistant professor with the Department of Computer Science and Engineering, Kangwon National University. His research interests include the design and analysis of computer networking and wireless communication systems especially for future Internet architectures.