Content-Centric and Named-Data Networking Security: The Good, The Bad and The Rest

Paolo Gasti
CS Dept., NYIT
Email: pgasti@nyit.edu

Gene Tsudik
CS Dept., UC Irvine
Email: gene.tsudik@uci.edu

***Abstract*—Named Data Networking and Content-Centric Networking (NDN and CCN, respectively) are closely related networking architectures which, unlike host-centric IP, emphasize content by explicitly naming it, and by making content names addressable and routable in the network. They support in-network (router-side) content caching, thus facilitating efficient and scalable content distribution, for which IP is comparatively poorly suited. These architectures also include new network-layer security features, such as signed content. While avoiding certain security problems of today's Internet, NDN and CCN trigger some new security and privacy issues. This paper overviews the security landscape of NDN/CCN, and focuses on two main areas of concern: (1) Interest Flooding Attacks, and (2) Producer, Consumer, and Content Privacy. We argue that, despite many attempts to fix these problems, they have not been fully addressed, and discuss the challenges that inhibit comprehensive solutions.**

I. Introduction

Despite its unprecedented and long-lasting success, today's Internet shows signs of age. Factors contributing to its aging include sheer scale of access, heterogeneity of devices, mobility, intermittent connectivity, and changing applications, all of which cause profound shifts in communication patterns. While there is no expected expiration date and no looming disaster, the Internet Protocol (IP), which forms the foundation of today's Internet, is widely considered to be in its sunset stage.

However, transitioning to a new Internet architecture is a monumental task. Indeed, it is hard to overestimate the difficulty of globally replacing IP. A perfect example of this is IPv6: despite being released almost two decades ago, its limited adoption [1] demonstrates that even gradual deployment of a new version of IP, which is far more modest than migrating to a brand new architecture, encounters major resistance and very long delays.

The research community, supported by (mostly) government and (to a lesser extent) industry funding, has long been aware of the need to re-think the Internet. To this end, several prominent architectures have been designed as potential successors to IP, sponsored by the US National Science Foundation (NSF) under the Future Internet Architectures (FIA) program. They include: XIA, MobilityFirst, NDN/CCN, and Nebula. European research efforts have resulted in NetInf, PSIRP/Pursuit, COMET, and GreenICN. Each comes with its own sets of novel features, benefits, and pitfalls. It is still very unclear which (if any) of them will ever replace IP. This uncertainty does not prevent or inhibit research activities related to these new architectures, as confirmed by numerous papers, reports, and software.

Of the four aforementioned architectures, NDN/CCN stands out. First, it is the only architecture that generated a set of focused venues, including: the yearly ACM ICN conference; workshops at Infocom, Globecom, and ICC; many journal special issues; and Dagstuhl seminars.[1] Second, only NDN/CCN managed to attract hundreds, if not thousands, of researchers world-wide. Third, it has its own very active IRTF working group—the Information-Centric Networking Working Group (ICNRG) with industry participation by prominent players, such as Cisco, Intel, and Ericsson.

Similar to its competitors, NDN/CCN was supposed to provide *"Security and Privacy by Design"*. This lofty goal was a key requirement set by the NSF. This paper focuses on major open security issues in NDN/CCN. Although NDN/CCN obviates certain security issues in IP and offers some interesting novel security features, it also triggers new security and privacy concerns. We believe that some of these can be (and have been) addressed, while others are more fundamental and either cannot be addressed at all, or require major architectural changes. Therefore, it is useful to highlight these tougher problems in order to attract more attention and motivate further research efforts.

Organization: Section II overviews NDN/CCN. Security and privacy issues that have already been successfully addressed are discussed in Section III. Sections IV and V focus, respectively, on outstanding security and privacy issues. The paper concludes in Section VI.

II. NDN/CCN Overview

This section summarizes key features of NDN/CCN. First, a note on nomenclature:

> *Named Data Networking (NDN) refers to the NSF FIA-funded project and the NDN codebase. Content-Centric Networking (CCN) originally referred to a broader notion of networking that, as the name suggests, focused on content. In that sense, NDN is an instance of CCN. However, in later years, CCN or CCNx denoted a separate research effort by Xerox PARC, which was acquired by Cisco in late 2016. Even though CCN and NDN diverged around 2013, they resemble each other very closely in terms of key features, such as: types of entities, processing of messages and message types. The differences are mostly minor; therefore, we use NDN/CCN as shorthand for both.*

[1] Although many of these venues cover the broader ICN concept, it is easy to see from programs and proceedings that NDN/CCN-related topics dominate.

In contrast to today's IP-based Internet architecture, which focuses on communication between end-points (i.e., interfaces/hosts and their addresses), NDN/CCN focuses on content by making it named, addressable, and routable. Moreover, each content must be signed by its producer. A content name is a URI-like string composed of one or more variable-length name segments, separated by the '/' character. To obtain content, an end-user (consumer) issues an explicit request message, called an *interest*, containing the name of desired content. This interest can be *satisfied* by the correspondingly named content found: (1) in a router cache, (2) at the content producer, or (3) at a repository (such as a CDN node) designated by the producer. Once found, the desired content is returned to the consumer along the reverse path of the preceding interest. Name matching is exact, e.g., an interest for `/us/edu/ucla/eng/cs/fileA` can only be satisfied by a content object with the very same name (or name prefix).

In addition to its actual payload, a content object includes several fields such as content name and content signature. The signature is computed over the entire content object, including its name. An interest carries a name, optional payload, and other fields that restrict the content response. Messages are moved within the network by routers. Each router has two mandatory, and one optional, components:

- *Forwarding Interest Base* (FIB): table of name prefixes and corresponding outgoing interfaces, used to forward interests based on longest-prefix-matching of their names.
- *Pending Interest Table* (PIT): table of outstanding (pending) interests and corresponding (incoming) interfaces.
- *Content Store* (CS): optional storage used for content caching. Henceforth, we use the term *cache* to refer to *CS*.

A router uses its FIB to forward interests towards the nearest copy of requested content. Whereas, a router uses its PIT to forward content along the reverse path (of the preceding interest) towards consumers. Specifically, upon receiving an interest, a router first checks its cache (if present) to see if it can satisfy this interest locally. In case of a cache miss, it checks its PIT for an outstanding version of the same interest. If there is a PIT match, the new interest's incoming interface is added to the PIT entry and the interest is discarded. (This is called *interest collapsing*.) Otherwise, a router creates a new PIT entry and forwards the interest to the next hop according to its FIB. For each forwarded interest, a router creates a new PIT entry containing the name in the interest and the interface from which it arrived, such that content may be returned to the consumer. When content is returned, a routers forwards it out on all interfaces listed in the matching PIT entry, and then removes the entry. A content that does not match any PIT entry is discarded.

III. NDN/CCN Security Issues

This section provides a high-level overview of security issues in NDN/CCN. We begin with a coarse-grained discussion of entity security, and then proceed to specific issues.

A. Entity Security

Host Security: Unlike IP, NDN/CCN does not have an explicit notion of a host or an end-system. However, hosts exist implicitly by playing consumer and/or producer roles. A host acts as a consumer when it issues an interest, and as a producer when it generates content.

A pure consumer-host, i.e., one that never produces any content, does not exist as an addressable entity. It thus has no assigned namespace and no corresponding public key that is used to verify its content. Consequently, routers are not supposed to forward interests to it. Furthermore, routers only forward content towards consumer-hosts that that have explicitly requested it. Therefore, a consumer-host should never receive unsolicited traffic from outside its broadcast domain. This is a clear and definite security advantage of NDN/CCN over IP.[2]

In contrast, in order to be able to receive interests, a producer-host needs to advertise its namespace. The same ability to receive incoming interests is also a means of receiving spurious interests, which leads to *Interest Flooding Attacks*, discussed in Section IV. For now, suffice it to say that a producer-host is essentially as (in)secure as an IP host on today's Internet.

Router Security As described in Section II, an NDN/CCN router is substantially more complex that its IP counterpart. The latter is basically stateless with respect to data traffic.[3] Whereas, an NDN/CCN router needs to maintain a PIT and (optionally) a cache. These two types of new state are directly influenced by hosts: consumers and producers. Also, both cache and PIT require specialized software support not present in IP routers. Moreover, an NDN/CCN router must be capable of (though not required to) verifying content signatures. This calls for additional cryptographic software and possibly hardware. Because of these additional complexities, NDN/CCN routers are subject to attacks that do not apply to IP routers. One such attack is discussed in Section IV.

B. Specific Security Issues

Cache and Content Poisoning: Cache poisoning involves injecting fake (generated by an incorrect producer) or corrupted (i.e., carrying an invalid signature) content into router caches [2]. Similarly, content poisoning involves injecting fake or corrupted content into the network [3]. The goal of these attacks is to increase content delivery cost for consumers and for the network.

In principle, cache and content poisoning can be addressed by requiring routers to verify signatures on the content they cache and/or forward. However, mandatory in-network signature verification raises major efficiency and trust management issues. The former, because signature verification is a computationally expensive operation, and the latter, because signature verification is meaningful only if routers *have* and *trust* the public key used to verify a signature. In practice, routers cannot be

[2]A malicious consumer-facing router can always flood a consumer-host with unsolicited traffic. However, this is not unique to NDN/CCN.

[3]The only soft state in an IP router is the FIB, which is influenced exclusively by control traffic.

expected to retrieve and validate one or more public keys for each content they forward or cache.

To address these issues, techniques such as randomized packet verification [2], signed catalogs [4], self-certifying names [5], and secure binding of namespaces and cryptographic keys [6], [7], [8] have been proposed.

Content Access Control and Cache Privacy: Because content can be stored in untrusted in-network caches, enforcing access control at individual routers is impractical. For this reason, NDN/CCN implements access control using content encryption [9]. To allow access to a particular piece of content, the producer must share the decryption key with all intended recipients. Any key management technique, such as proxy re-encryption [10], [11] and attribute-based encryption [12] can be used to implement flexible and fine-grained access control.

As discussed in Section V, content encryption does not guarantee user privacy. Moreover, content encryption does not prevent the adversary from learning whether a particular content object—encrypted or otherwise—has been cached. This is a significant threat to the privacy of producers [13] and consumers [13], [14]. To improve cache privacy, techniques based on concealing cache hits [15], [16] and anomaly detection [17] have been proposed.

Anonymous Communication: In contrast with IP packets, NDN/CCN interests do not include a sender identifier. While this might seem to offer better consumer anonymity, Ambrosin et al. [18] showed that it is possible to identify which consumer issued a particular interest by exploiting in-network cache. Onion routing protocols specific to NDN/CCN have been proposed as a way to implement anonymous communication [19], [20]. However, similarly to onion routing in IP, these techniques result in increased latency and reduced bandwidth. As such, they are not intended for carrying a substantial portion of Internet traffic. In addition, onion routing takes away the benefits of in-network caching.

Other Issues: Due to space limitations, we do not consider certain other security and privacy issues in NDN/CCN that are somewhat more peripheral or less urgent than those discussed above, such as secure routing, network-layer trust, cache pollution, and security for specialized (e.g. IoT) network settings.

IV. Interest Flooding Attacks

Interest-Flooding Attacks (IFAs) are the bane of NDN/CCN security. A successful IFA is relatively easy to implement and its impact can be devastating. The starting point is a botnet composed of potentially many topologically distributed zombies, i.e., compromised consumer-hosts. This requires no leap of faith, given recent examples of such botnets, e.g., Mirai [21]. On cue from a Command and Control Center (CCC), the botnet targets a set of routers and/or producers by generating large numbers of closely spaced *pseudo-interests*. Each such pseudo-interest carries a content name that starts with legitimate and routable prefix of an existing producer, and terminates with a random (or otherwise non-sensical) suffix, e.g., `/us/edu/MIT/eecs/web/news/`*`random-suffix`*.

Because successfully mitigating IFA is challenging, if at all possible (see Section IV-A below), and because IP routers are not susceptible to this attack (they do not maintain user-initiated state for the purpose of packet forwarding), we consider IFA a major problem for NDN/CCN. Below we overview IFA, and discuss why current technique do not fully address it. We partition overall impact of a pseudo-interest into three categories: (1) router processing, (2) router state, and (3) producer processing.

Router Processing: Each router that receives a pseudo-interest must perform four tasks, not necessarily in this order: (1) PIT lookup, (2) cache lookup, (3) FIB lookup, and (4) new PIT entry insertion. Task (1) is very fast, especially if hash tables and/or associative memory is used; its cost can be made nearly constant. Whereas, task (2) is potentially more expensive because a cache likely contains orders of magnitude more items than a PIT. A FIB lookup is also relatively expensive, mainly due to longest-prefix matching, whereby successive concentric prefixes of a name are hashed and separately looked up in a FIB, in order to determine the most specific (longest-prefix) entry. Finally, a PIT insertion is usually faster than (2) and (3), though it is unlikely to cost less than (1) [22].

Indeed, a normal (non-malicious) interest can also cause a router to perform all four aforementioned tasks. However, if a prior interest for the same name is already pending (i.e., a PIT entry exists), PIT collapsing takes place, and steps (2), (3), and (4) need not be performed. Similarly, a cache lookup might result in a hit, in which case (3) and (4) can be avoided. In contrast, *every* pseudo-interest requires *each router* to perform all four tasks.

Router State: A barrage of closely spaced pseudo-interests can quickly fill up a router's PIT. A full PIT leaves a router with few options, such as: (1) drop new interests; or (2) randomly delete current entries; or (3) delete oldest entries before they expire. Regardless of the strategy, sustained IFA can essentially squeeze out legitimate traffic and result in the complete abuse of the PIT. Furthermore, each pseudo-interest leaves lingering state in each router in the form of a new PIT entry. Such an entry is never removed as a result of the arrival of a matching content. Instead, it simply expires. If expired PIT entries need to be explicitly removed, a router would wind up spending more cycles on that task.

Another negative impact can occur if a router decides to use multi-path forwarding of interests [23], [24]. This is generally viewed as a positive feature in NDN/CCN: a router can choose to forward an interest out on multiple interfaces if its FIB contains multiple equi-prefix entries for the same name. The positive rationale behind this feature is that, though seemingly wasteful, it may result in lower latency for the desired content. Though this may well hold for benign interests, pseudo-interests happily "benefit" from this feature as it amplifies their effect. (Further amplification would also occur if a router forwards interests close to their expiration to additional interfaces *sequentially* [25].)

Producer Processing: Because each pseudo-interest carries a

name with a random (and, with high probability, unique) suffix, it can neither be satisfied by a cache hit, nor be suppressed via PIT collapsing. Therefore, pseudo-interest will eventually arrive to a producer. The latter recognizes a prefix as one of its own and performs a cache lookup, which clearly results in a miss. Next, the producer passes the pseudo-interest to the application, which likely determines that this interest is junk and discards it. Unfortunately, by this time, the producer has incurred costs associated to the bandwidth used by the interest, a cache lookup, and layer switching. The resulting impact is similar to that of DoS/DDoS attacks on today's IP-based hosts.

A. IFA Countermeasures

We now overview several types of IFA mitigation techniques.

Simple Techniques: Given a router total bandwidth and the router's PIT expiration timeout, it is possible to determine the minimum PIT size necessary to prevent the router from succumbing to a sustained IFA. However, deploying PITs that are both large enough *and* fast enough is both wasteful (most PIT entries would normally be empty), and probably infeasible with current technology [26].

Alternatively, a router could set a small (e.g., sub-second) PIT timeout, thus increasing the cost of flooding its PIT. As a downside, legitimate interests will expire often. However, since PIT timeout does not affect an interest's probability of reaching the nearest copy of desired content, each subsequent interest retransmission would likely disseminate the content in caches closer to the consumer, thus eventually enabling content retrieval. As a result, reducing PIT timeout shifts state from routers to consumers, because the latter are responsible for keeping track of expired interests. Moreover, it trades off PIT state for bandwidth used by interest retransmission. The net benefits of this approach are therefore unclear.

Anomaly/Attack Detection: The primary goal of IFA countermeasures based on anomaly detection is early attack identification. Once an attack is identified, routers can take countermeasures, such as alerting neighbors, rate-limiting (aka throttling) specific interfaces and/or namespaces, or a combination thereof. Detection can occur at a single router [27], [28], [29], or as a result of cooperation among multiple routers [27], [29]. Both [29] and [27] use various PIT statistics to detect attacks, e.g., ratio of timed-out interests and PIT utilization for each interface, each FIB entry and each name prefix.

For these techniques to succeed, attack detection must be quick and inexpensive, and reaction—swift, fair, and effective. Furthermore, cooperative detection techniques must incur minimal extra communication overhead for routers. Unfortunately, no current technique satisfies these requirements: unconstrained network anomaly detection is, in general, a fundamentally hard and largely unsolved problem in the context of network security [30], [31]. Therefore, the entire notion of deployability of NDN/CCN comes into question if *routers* (rather than end hosts or dedicate security devices) must either perform reliable anomaly detection at wire speed, or succumb to IFA.

Although aforementioned reactive techniques attempt to be fair, they end up penalizing both adversarial pseudo-interests and legitimate consumers' genuine interests. Thus, the adversary might be able to exploit router reactions to *amplify* the attack. However, on the positive side, these techniques aim to discard, as soon as possible, suspected pseudo-interests. If successful, this in principle mitigates all IFA impact factors: router processing, router state, and producer processing. (Albeit, at the cost of extra router state in the form of various PIT statistics.)

PIT-less Routing: Another way to mitigate IFA is to replace NDN's stateful forwarding plane with a stateless one, thus eliminating the PIT. Ghali et al. [32] showed how to implement PIT-less NDN/CCN routers by including a *backward routable name* (BRN) in each interest. The entity that satisfies an interest appends the BRN to the content, which is then forwarded back to the consumer the same way that an interest is forwarded. In other words, a router uses LPM-based BRN lookup in its FIB to determine the interface(s) on which to forward the content back to the consumer. This way, an interest and a corresponding content can take different paths. Also, a consumer needs to have its own routable name (akin to a producer); this sacrifices consumer opaqueness that we identified in Section III-A as one of the major benefits of NDN/CCN.

Alston et al. [33] proposed an alternative PIT-less routing technique that involves putting PIT state "on the wire". In it, a router that receives an interest looks up its cache and, in the event of a cache miss, creates a structure similar to a new PIT entry. However, this structure is appended the interest and forwarded, instead of being stored locally. An interest thus accumulates such exported PIT entries before eventually reaching an entity that has the desired content. The latter forwards these PIT entries along with the content. Each intervening router uses its own entry to forward the content towards the original consumer. This scheme involves growing interests and shrinking content messages.

While seemingly appealing, both PIT-less techniques have considerable downsides. Without a PIT, a router cannot aggregate outstanding interests requesting the same content; it has to forward all interests. This reduces NDN's ability to carry multicast traffic efficiently. Although lack of interest aggregation might not significantly increase latency or bandwidth consumption of content, it incurs higher router processing cost and bandwidth for interests. Furthermore, in [32] forwarding decisions for interest and content objects are performed independently, thus twice as many FIB lookups are needed for each interest/content pair. (Normally, content processing in routers does not involve any FIB lookups). Also, lack of PIT deprives routers of information on pending requests. This information is useful for monitoring and improving NDN/CCN forwarding performance [23], [24].

By avoiding the PIT, PIT-less techniques mitigate IFA router state and processing impacts. Unfortunately, they do nothing to address producer processing, i.e., a producer can still be flooded by pseudo-interests. It is easy to see that, if all routers adopt either PIT-less scheme, effects of IFA on producers would be equivalent to DDoS attacks in today's Internet.

B. Tentative Future Directions

Given limited efficacy of current IFA mitigation techniques, developing more substantial countermeasures would require exploring new directions, such as *hybrid PIT-less forwarding* that we sketch out in this section.

With hybrid PIT-less forwarding, a router uses its PIT as in current NDN/CCN until it fills up. At that point, instead of dropping a new interest, a router appends PIT state to it before forwarding to the next hop(s), as described in [33]. This process can be reversed any time by any router with free space in its PIT: upon receiving an interest that carries PIT state, the router inserts that state to its PIT, and forwards the interest *without* the forwarding state. As a result, adding and removing PIT state from interests may happen multiple times on the interest path to the producer because of different level of PIT congestion in different parts of the network.

The benefits of this approach are: (1) in contrast with [32] and [33], routers can take advantage of information in their PITs to make better forwarding decisions, until a PIT fills up; (2) communication overhead of interests and content is, on average, less than that of [33], since forwarding information is added only by *some* routers involved in forwarding the interest, and can be removed by one or more upstream routers; (3) when this technique is used in conjunction with anomaly detection, detection latency and reaction effectiveness are not as critical as with anomaly detection alone, since failure to detect (or to react to) IFA does not prevent routers from forwarding interests; and (4) in contrast with [32], forwarding content to consumers requires routers to make the same forwarding decisions as in current NDN/CCN routers.

In summary, the worst-case scenario under the hybrid technique is analogous to the normal state of PIT-less approaches, and significantly better than the worst case of anomaly detection, wherein almost no interest are forwarded. Clearly, costs and benefits of the hybrid technique need to be carefully evaluated to determine the extent to which it really mitigates IFA.

V. Producer, Consumer, and Content Privacy

Despite extensive amount of work on access control, cache privacy, name encryption, and anonymous communication, privacy remains a major outstanding problem in NDN/CCN. Given widespread adoption of encrypted communication protocols,[4] and recent push for on-line user privacy [35], NDN/CCN should provide no less privacy than today's TLS [36].

TLS implements end-to-end confidentiality, with recent versions of the protocol supporting forward secrecy. Because TLS encrypts traffic above the network layer, the IP addresses of the hosts communicating via TLS can be observed by the adversary. Also, DNS queries made by the client prior to establishing a TLS connection leak substantial information.[5] While hiding the endpoint IP addresses from most observers requires fairly expensive tools (e.g., Tor, VPNs), the content of DNS queries and responses can be concealed using efficient protocols such as DNS Over HTTPS (DoH) [37]. This is important, because a single IP address might host multiple unrelated domains (e.g., with shared hosting providers and CDNs), and therefore IP addresses alone do not disclose which website is being accessed. As a result, TLS and DoH provide a reasonably strong level of privacy to users, while imposing limited computation and bandwidth overhead.

Unfortunately, NDN/CCN appears to be unable to provide the same level of privacy as the current Internet. In what follows, we identify two source of information leakage: content names, and content objects. We believe that this leakage results from a privacy/efficiency tradeoff that is intrinsic to the design of NDN/CCN. As such, it might be impossible for NDN/CCN to simultaneously offer strong privacy *and* efficient forwarding, even just to the same level as TLS and DoH.

Name Privacy: NDN/CCN content names disclose a significant amount of information through their routable and non-routable components. Routable components are analogous to a combination of current DNS names and IP addresses: they are familiar human-readable strings (as in DNS names) that are used by routers to identify the next hop for interests (similar to IP addresses). Naturally, routable name components can be hashed, encrypted, or can be replaced with random strings (without loss of generality, in what follows we refer to all these techniques as *encrypted names*). However, in addition to be not human-readable, encrypted routable name components provide very limited privacy. To keep a FIB to a manageable size, and to take advantage of router caches and interest collapsing, encryption of each routable name component must be effectively deterministic. This implies that the adversary can easily correlate different interests (some of which could be generated by the adversary itself) based on the use of the same routable name components. As a result, information leaked by the routable part of NDN/CCN names is analogous to what is currently leaked by DNS queries.

Non-routable components allow more flexibility in terms of encryption. Each component, or sequence of components, can for instance be encrypted independently by each consumer under the producer's public key using a probabilistic scheme, thus providing strong privacy guarantees. However, as a result, content cannot be satisfied using in-network caches (except for retransmission due to packet loss), and interests cannot be collapsed in PITs, because no two interests for the same content have the same name. Furthermore, producers must individually sign *each* content object requested by *each* consumer, because the signature on a content object covers its name, which must match the name in the interest that requested it. This adds a considerable cost for producers compared to TLS, where *public-key* cryptographic operations are performed only at the beginning of a new connection, rather than for each packet. These drawbacks could be addressed using deterministic encryption, hash functions, or fixed (pseudorandom) strings to

[4] As of April 2018, 69% of all pages accessed using Firefox are retrieved via SSL/TLS [34].

[5] In this discussion, we ignore side-channel attacks that leverage packet delay and size, because similar attacks are likely possible under most network architectures, including NDN/CCN.

encode non-routable components. However, this would result in the same weak privacy properties associated with encrypted routable name components.

Onion routing techniques, such as Andana [19] and AC^{3}N [20], can mitigate these issues. However, exit nodes can still observe full content names. This is a problem, because in Tor exit nodes have been abused to intercept traffic [38], and it is reasonable to expect the same to happen with Andana and AC^{3}N.

Content Privacy: As with name privacy, there is a tension between efficient content distribution and content privacy. As discussed in [9], content can be encrypted once by its producer, and the corresponding decryption key can be shared with all intended consumers. The downside of this approach is that the adversary can easily determine which consumers are accessing a particular encrypted content object, thus linking users with similar interests. This leaks substantially more information than TLS, where the adversary cannot determine whether information exchanged as part of two TLS connections overlaps. Also, forward secrecy (which TLS supports) is unattainable if content is encrypted once for all consumers. As with name privacy, onion routing only partially mitigates this issue.

A more privacy-friendly approach would require producers to encrypt content individually for each consumer.[6] However, encrypted traffic would not benefit from caching and interest collapsing. It would also impose additional signing overhead on producers, since each encrypted copy of the same content object would have to be signed individually.

To summarize, as far as leakage of sensitive information, NDN/CCN is potentially a substantial step backwards with respect to privacy, and possibly performance, as compared to IP-with-TLS. Of course, IP-with-TLS can be used as an overlay over NDN/CCN. However, this should be considered at best a stop-gap measure, rather than a long-term way to address privacy in NDN/CCN.

VI. Conclusion

In this paper, we reviewed the current landscape of NDN/CCN security and privacy. We showed that NDN/CCN addresses many security issues as well as – and, in some cases, better than – IP. However, we also claim that NDN/CCN has two important unsolved problems: (1) Interest Flooding Attacks, and (2) User and Content Privacy. Despite many attempts, they not been fully addressed. More importantly, comprehensive solutions to these problems appear to be fundamentally at odds with NDN/CCN core features and design choices. We hope that highlighting these problems will stimulate new research efforts aiming to address them.

References

[1] "State of ipv6 deployment 2017," https://www.internetsociety.org/resources/doc/2017/state-of-ipv6-deployment-2017.

[2] P. Gasti *et al.*, "Dos and ddos in named data networking," in *ICCCN*. IEEE, 2013.

[6]To ensure that interests are satisfied exclusively by a producer, a consumer can append a random component at the end of the name in an interest.

[3] C. Ghali *et al.*, "Needle in a haystack: Mitigating content poisoning in named-data networking," in *NDSS Workshop SENT*, 2014.

[4] M. Baugher *et al.*, "Self-verifying names for read-only named data," in *INFOCOM Workshops*. IEEE, 2012.

[5] C. Ghali *et al.*, "Network-layer trust in named-data networking," *ACM CCR*, 2014.

[6] X. Zhang *et al.*, "Towards name-based trust and security for content-centric network," in *ICNP*. IEEE, 2011.

[7] B. Hamdane *et al.*, "Named-data security scheme for named data networking," in *Network of the Future (NOF)*. IEEE, 2012.

[8] A. Afanasyev *et al.*, "Snamp: Secure namespace mapping to scale ndn forwarding," in *INFOCOM Workshops*. IEEE, 2015.

[9] D. Smetters *et al.*, "Securing network content," PARC, Tech. Rep., 2009.

[10] C. A. Wood *et al.*, "Flexible end-to-end content security in ccn," in *CCNC*. IEEE, 2014.

[11] R. S. da Silva *et al.*, "An access control mechanism to ensure privacy in named data networking using attribute-based encryption with immediate revocation of privileges," in *CCNC*. IEEE, 2015.

[12] M. Ion *et al.*, "Toward content-centric privacy in icn: Attribute-based encryption and routing," in *SIGCOMM workshop on ICN*. ACM, 2013.

[13] G. Acs *et al.*, "Cache privacy in named-data networking," in *ICDCS*. IEEE, 2013.

[14] A. Compagno *et al.*, "Violating consumer anonymity: Geo-locating nodes in named data networking," in *ACNS*. Springer, 2015.

[15] G. Acs *et al.*, "Privacy-aware caching in information-centric networking," *IEEE Transactions on Dependable and Secure Computing (TDPS)*, 2017.

[16] A. Mohaisen *et al.*, "Protecting access privacy of cached contents in information centric networks," in *ASIACCS*. ACM, 2013.

[17] M. Gao *et al.*, "Protecting router cache privacy in named data networking," in *ICCC*. IEEE, 2015.

[18] M. Ambrosin *et al.*, "Covert ephemeral communication in named data networking," in *AsiaCCS*, 2014.

[19] S. DiBenedetto *et al.*, "ANDaNA: Anonymous named data networking application," in *NDSS*, 2012.

[20] G. Tsudik *et al.*, "Ac3n: Anonymous communication in content-centric networking," in *CCNC*. IEEE, 2016.

[21] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *USENIX Security Symposium*, 2017.

[22] H. Yuan *et al.*, "Scalable ndn forwarding: Concepts, issues and principles," in *ICCCN*. IEEE, 2012.

[23] A. Udugama *et al.*, "An on-demand multi-path interest forwarding strategy for content retrievals in ccn," in *NOMS*. IEEE, 2014.

[24] L. Wang *et al.*, "Ospfn: An ospf based routing protocol for named data networking," Technical Report NDN-0003, Tech. Rep., 2012.

[25] L. Zhang *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[26] D. Perino *et al.*, "A reality check for content centric networking," in *SIGCOMM workshop on ICN*. ACM, 2011.

[27] A. Compagno *et al.*, "Poseidon: Mitigating Interest Flooding DDoS Attacks in Named Data Networking," in *LCN*, 2013.

[28] H. Dai *et al.*, "Mitigate ddos attacks in ndn by interest traceback," in *INFOCOM Workshops*. IEEE, 2013.

[29] A. Afanasyev *et al.*, "Interest flooding attack and countermeasures in named data networking," in *IFIP Networking*. IEEE, 2013.

[30] S. Axelsson *et al.*, "The base-rate fallacy and the difficulty of intrusion detection," *TISSEC*, 2000.

[31] M. H. Bhuyan *et al.*, "Network anomaly detection: methods, systems and tools," *IEEE communications surveys & tutorials*, 2014.

[32] C. Ghali *et al.*, "Living in a pit-less world: A case against stateful forwarding in content-centric networking," *arXiv preprint*, 2015.

[33] A. Alston *et al.*, "Neutralizing interest flooding attacks in named data networks using cryptographic route tokens," in *NCA*. IEEE, 2016.

[34] "Let's encrypt stats," https://letsencrypt.org/stats/.

[35] "Home page of EU GDPR," https://www.eugdpr.org.

[36] T. Dierks *et al.*, "The TLS protocol version 1.2," *IETF*, 2008.

[37] "DNS over HTTPS (DoH)," https://datatracker.ietf.org/wg/doh/about/.

[38] P. Winter *et al.*, "Spoiled onions: Exposing malicious Tor exit relays," in *PETS*. Springer, 2014.