Security and Privacy in Future Internet Architectures

Benefits and Challenges of Content Centric Networks

Roman Lutz
College of Information and Computer Sciences
University of Massachusetts Amherst
romanlutz@cs.umass.edu

ABSTRACT

As the shortcomings of our current Internet become more and more obvious, researchers have started creating alternative approaches for the Internet of the future. Their design goals are mainly content-orientation, security, support for mobility and cloud computing. The probably most popular architecture is called Content Centric Networking. Every communication is treated as a distribution of content and caches are used within the network to improve the effectiveness. While the performance gain of Content Centric Networks is undoubted, there are questions about security and especially privacy since it is not one of its main design principle. In this work, we compare the Content Centric Networking approach with the current Internet with respect to security and privacy. We analyze improvements that have been made and new problems that have yet to be resolved. The Internet of the future could be content-oriented, so it is essential to identify potential security and privacy issues that are inherent to the architecture early on.

Categories and Subject Descriptors

C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Store and Forward Networks*

General Terms

Security

Keywords

Future Internet Architecture, Content Centric Networking, Privacy, Censorship Circumvention

1. INTRODUCTION

Since the beginnings of the Internet, the world has changed a lot. Back then, all computers were stationary and used mainly by scientists who trusted each other. The computers themselves had scarce resources and their Internet connections were very slow. The situation is mirrored in the paper by David Clark [20] from 1988 in which he describes the design philosophy of the current Internet. The top goals include fault tolerance, flexibility to allow different kinds of communications and inclusion of a variety of networks. Regardless of the extent to which these and other goals were achieved, we can conclude that the main design principles do not correspond to today's requirements. As a consequence, it is not surprising that we can observe a lack of security, privacy, support for mobility and efficiency among other shortcomings. Even though a lot of work has focused on adding these desired features, e.g. with encryption and new protocols to handle mobility, inherent problems of the architecture can only be overcome to a certain degree. For many years, researchers all over the world have tried to find new clean-slate architectures to replace the current Internet at some point in the future. Specifically, the NSF-funded projects Future Internet Design (FIND), Future Internet Architecture (FIA) [4] and FIA - Next Phase (FIA-NP) aim at designing the next-generation Internet. After a multitude of ongoing sub-projects in the early phases, FIA-NP includes only the three most promising remaining approaches: MobilityFirst [2], eXpressive Internet Architecture (XIA) [5] and Named Data Networking (NDN) [3].

As the name suggests, MobilityFirst focuses on support for mobility at a time when the majority of devices are mobile. It uses a highly scalable Global Name System for name resolution. XIA aims at accommodating a variety of network services and being flexible enough for new innovations of the future. NDN is based on the Open Source project CCNx, which implements the idea of a Content Centric Network. Most communications on the Internet nowadays have the goal of retrieving content, and others can be modelled as an exchange of content. Since popular content is requested frequently, Content Centric Networks involve routers with integrated caches to improve the scalability.

All of the three projects state improved and inherent security as a primary objective. What they really mean is verifiable integrity of communications. Other security problems that already exist in the current Internet are not necessarily solved and there are even new security vulnerabilities. Apart from that, none of the research groups declared privacy as a main design objective. It is logical that this can have consequences for users.

In this paper, we focus on the content-oriented approach taken by NDN, the most popular project with the largest research community. We first describe the architecture of a

content-oriented Internet in Section 2. Afterwards, we compare the current Internet with Content Centric Networks with respect to security and privacy. Section 3 describes what remains unchanged before Sections 4 and 5 examine benefits and challenges of Content Centric Networking. Finally, Section 6 gives an overview of related research followed by a conclusion.

2. CONTENT CENTRIC NETWORKING

The idea behind Content Centric Networking is to adapt the messages sent over the Internet to what they really are: content. Instead of of the restriction to end-to-end communications between pairs of users, Content Centric Networking allows for much more flexible and efficient message exchange. The main principle is that everything is content. In contrast to the current Internet's *push* model, Content Centric Networking uses a *pull* communication model. If a host wants to see a specific piece of content, he can simply request it by its name. In both CCNx and NDN, content names are hierarchical, e.g. `/umass/cs/cs660/student/report`. In theory, flat naming could have been used instead, but that requires a name resolution. A discussion of benefits and problems of hierarchical and flat naming goes beyond the scope of this paper. The hierarchical content name is sent as a so-called *interest* to the next router. Routers communicate with each other by sending name prefixes that they can serve. This is similar to routing based on IP prefixes in the current Internet. Before forwarding the interest to the next router, every router will first check whether the queried content is in its cache. If no router has the content in its cache, the interest is forwarded all the way until it arrives at the content provider. He will send the name of the content, the content itself and his signature back. The signature is just the hash of the name and content, `h(name, content)`, encrypted with the provider's private key. It allows the recipient to verify data and source integrity. For that, he needs to hash the name and content himself, decrypt the signature with the provider's public key and compare the them. This verification step is necessary, because hosts will often receive content from a cache instead of the content provider. An example scenario is shown in Figures 1, 2, 3, 4 and 5.
Compared to the current Internet, routers are very different. They basically contain three tables: the cache, the Pending Interest Table (PIT) and the Forward Information Base (FIB). Based on their cache replacement policy routers decide to cache content that is forwarded through them or not. The PIT registers every interest that the router receives and the interface from where it arrived. By looking up the name in the FIB, which maps name prefixes to outgoing interfaces, the router finds out where to forward interests if they are not cached. The way the FIB is used is very similar to longest prefix matching of IP prefixes in routers of the current Internet. Routers communicate with each other to fill and update the FIB. Apart from reducing the network congestion by returning cached content, routers also aggregate interests. If multiple interests for the same content arrive before the router receives the content, it only forwards the first interests and keeps all requesting interfaces in the PIT. Once the content arrives, it is forwarded to all of the requesting interfaces. This illustrates how multicast routing is possible without additional effort. In general, routers drop all interests from the PIT after the content is forwarded.

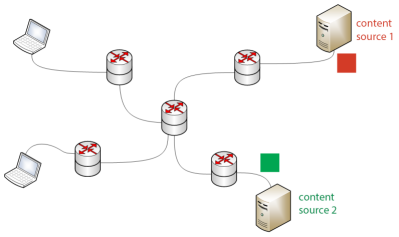

Figure 1: An example of Content Centric Networking[1] (1). Two hosts are on the left, two content providers on the right. The routers in the network have caches.

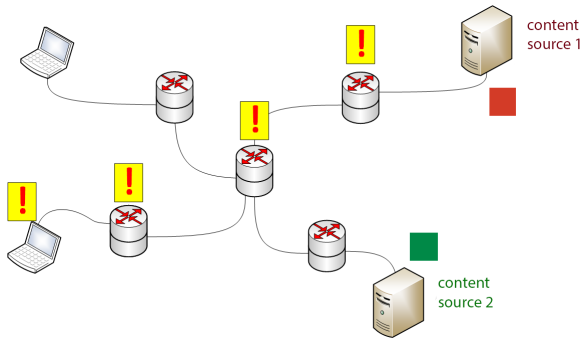

Figure 2: An example of Content Centric Networking[1] (2). The host in the bottom left corner requested content from the content provider in the top right corner. None of the routers between them had the content cached, so the interest is forwarded.

If the content does not arrive due to loss, the interest will be dropped after a timeout. In Sections 4 and 5, we will describe the consequences this design has on security and privacy.
The content itself can be sent in the clear or encrypted. In order to allow for scalable communication, the names should not be encrypted at least for popular content in order to allow for the use of caching. If the names are encrypted, the requesting user will not know the name for which to query unless he knows the secret encryption key. We discuss this issue in greater detail in Section 5.

3. UNCHANGED SECURITY AND PRIVACY COMPARED TO CURRENT INTERNET

Even though the architecture is different from the current Internet, some problems have not been solved. Several of them have to do with censorship. Censorship Circumvention - or more generally privacy - is not defined as a primary design principle of Content Centric Networks, so additional techniques will have to be built on top of the architecture to support it. Firstly, the content provider is still easily identifiable. While it is possible to do this through his IP address in the current Internet, Content Centric Networking

[1]The pictures used for this graphic are from wikimedia.org and wikipedia.org. They are referenced at the end of this document.

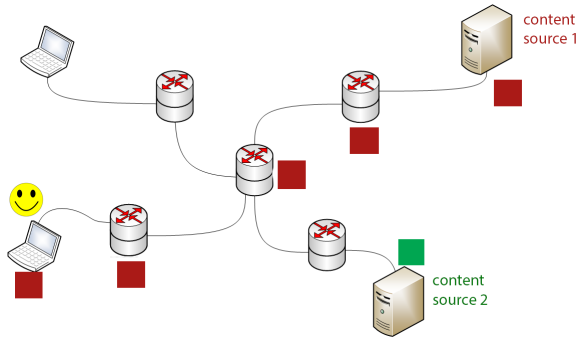

Figure 3: An example of Content Centric Networking[1] (3). The content provider returned the requested content and name signed with his signature. The content is forwarded backwards in the same path the interest came. The routers can cache the content or not dependent on their cache replacement policy.

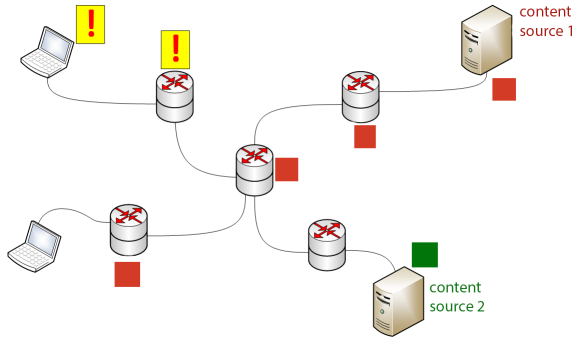

Figure 4: An example of Content Centric Networking[1] (4). Another user requests the same piece of content that is already stored close by in a router's cache.

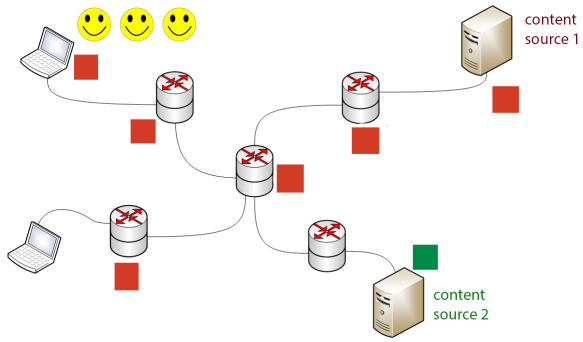

Figure 5: An example of Content Centric Networking[1] (5). The router recognizes the content name as cached content and returns it.

forces him to sign every piece of content with this signature. As a consequence, censors can check whether arriving content is coming from an unwanted source and possibly filter it. We will look into this problem in Section 5.4. Secondly, censors are currently able to block known Tor relays and proactively probe hosts in order to find out whether they are Tor bridges. If they suspect a host, they can simply behave as if they were a benign user and try to access Tor through the bridge. If it works, they have identified a Tor bridge. In Content Centric Networks, anonymity networks are also conceivable. Section 5.8 examines existing solutions such as AND$\bar{a}$NA. Similarly to Tor, AND$\bar{a}$NA's relays can be blocked by censors. To our knowledge, bridges have not been proposed yet, but they could be tested similarly to proactive probing for Tor bridges. Thirdly, censors can filter interests for banned keywords or prefixes in names. Further techniques like Deep Packet Inspection (DPI) are also applicable. Finally, trusted third parties like Certification Authorities (CA) in the current Internet will still be necessary. Even though every piece of content has to be signed by the publisher, the recipient needs a trusted third party delivering the public key in order to be able to verify the integrity of the content.

4. SECURITY AND PRIVACY BENEFITS OF CONTENT CENTRIC NETWORKS

In the following, we describe how the users' security and privacy benefit from the Content Centric Networking architecture. This includes verifiable integrity with signed content, the absence of addresses in interests, abandonment of name resolution in favor of hierarchical names and a certain degree of protection against Denial of Service (DoS) attacks.

4.1 Verifiable Integrity

As described in Section 2, it is mandatory for content providers to sign the content and name with their private key. Therefore, the recipient can verify the integrity of data and source. This prevents spoofing as it is possible in the current Internet. A positive side-effect of it is Routing Security. As every other piece of content, communication between routers has to be signed. This prevents adversaries from influencing the routing tables, i.e. the FIB in Content Centric Networks.

4.2 Absence of Addresses

Content Centric Networking abandons addresses totally. Neither interests nor the content delivery messages contain addresses. Interests in Content Centric Networks contain only the name of the requested content, but not who requested it. Only the first forwarding router knows the interface from which the content was requested. All other routers only know the previous router on the forwarding path. When the content provider returns the content, his message also includes the content name, his signature, the publisher's ID and information about where to retrieve the publisher's public key. As a result, there is no necessity for addresses, although the publisher can be inferred from the ID and the key. From a privacy perspective, this design improves anonymity because the source of an interest is unknown or at least hard to find out. In Section 5.2, we explain various techniques with which an adversary can still find out who

sent the interest.
But the different communication paradigm has even more advantages. In the current Internet, many attacks require the attacker to directly send messages to the victim. In Content Centric Networks the victim has no address, so this threat is completely mitigated because no content can arrive at a host without the host requesting the content in advance. It is slightly different for content publishers. This case will be examined in the next section.

4.3 Protection against Denial of Service

Apart from the protection against direct attacks on hosts, Content Centric Networking even offers a protection against Denial of Service (DoS) attacks on content providers. Assume an attacker launches a DoS attack on a publisher by sending lots of interests. These are simply collected at the first router and only one interest is forwarded. Even if the attacker controls a large botnet that is distributed over many locations that do not share routers on the routing path to the publisher, interest aggregation mitigates most of the attack. What actually happened is a shift in the point of attack from the content provider to the routers. Instead of flooding the publisher, the attack fills routers' Pending Interest Tables. We discuss this new DoS attack and possible coutermeasures in Section 5.1. Although neither NDN nor CCNx include this feature, some researchers have suggested adding a flag to interests that indicates that the interest should be routed to the content provider without returning cached content. Such a flag allows attackers to directly flood the publishers again.

4.4 Name Resolution

Content Centric Networks are based on hierarchical names and therefore do not require explicit name resolution. In the current Internet, name resolution is provided by DNS and offers a multitude of possibilities for attackers to exploit, e.g. DNS hijacking. If flat names are used instead, the advantage is lost.

5. SECURITY AND PRIVACY CHALLENGES OF CONTENT CENTRIC NETWORKS

Even though there are security and privacy aspects that improve with Content Centric Networking, it also poses new challenges. In this section, we outline several of these concerns and suggest possible solutions. We start with Interest Flooding Attacks, the most studied attack on Content Centric Networks and go on to describe less known and privacy-related problems.

5.1 Interest Flooding Attacks

As explained in Section 4.3, DoS attacks in Content Centric Networks will target routers instead of content providers. For that reason, they are called *Interest Flooding Attacks* (IFA). They have been the subject of many research papers over the last few years. In this section, we describe the problem in general and a few simple countermeasures. For an overview of more sophisticated countermeasures, we refer to Section 6. To carry out an IFA, an attacker needs to sends lots of interests. If the interests are all for the same piece of content, the first router aggregates the interests in its PIT. As soon as the number of pending interests surpasses the PIT capacity, the router drops or rejects interests. The denial of service on the first router works at least until the content actually arrives at the router and the requests can be answered. Effectively, the attacker just degraded his own local network's performance by flooding the router. If a router closer to the core or close to a specific content provider is the target, the attacker adapts the attack to include various different interests for content by the same publisher. Using different content names helps against interest aggregation. Ideally, the attacker controls a botnet that is distributed over many places, such that interest aggregation does not take place too early. Some routers then become bottlenecks for interests, resulting in the same situation as before, albeit at different routers. The situation becomes even more troubling if the attacker controls a remote content source. He could make the source answer very slowly on purpose to delay the content, while at the same time filling up the PIT with interests.
There is a relatively simple countermeasure used in NDN. Routers could keep track of the number of interests per domain and drop some of them if there are too many in one specific domain. Against an IFA, this is an effective strategy, but also has some issues. If there is a very popular event, e.g. the Superbowl, that most people watch, and the router starts dropping the interests, this is actually counterproductive. Of course, this also applies to all kinds of popular content. In fact, it allows the attacker to selectively block certain content by sending lots of requests. Most of his interests are dropped, but the same applies to legitimate hosts.
Depending on the configuration of the routers, they might verify the integrity of the content themselves. This involves expensive operations that can slow down the router. Especially in a scenario with a botnet and a cooperating content provider, the routers' speed could be reduced. In order to handle high load situations, routers could stop verification as soon as it becomes a burden for their forwarding speed.
Similarly, an adversary could try to degrade the network performance by filling the cache with arbitrary data. This is called *cache pollution*. Again, this works best with a cooperating content provider delivering the content and a botnet requesting it. The attack basically works with any kind of content that is not of any use to other hosts. Such a scenario is detectable by comparing cache hit rates. If only the same people request the content over and over again and nobody else is interested in it, it is very likely a cache pollution attack. It is more difficult to defend against such an attack. Countermeasures include blacklisting the cooperating source provider such that its content is not cached any more and ignoring the interests of recognized bots for caching or even completely.

5.2 Cache Privacy

The main privacy concern in Content Centric Networks is caching. Having a copy of my communications stored in caches that are available to everybody who knows how to query it is a risk. In the current Internet, adversaries have to sniff on traffic exactly at the time it is sent, because there are supposed to be no traces of the actual content. This is substantially harder than being allowed to query the messages by name and even some time after they were sent. As a consequence, the use of caches causes a tradeoff between privacy and efficiency. In the following, we will examine at-

tacks and possible countermeasures.
There are basically three attack scenarios as identified by Lauinger [40] and Acs et al. [7]:

- Cache Enumeration
- Timing Attack
- Conversation Cloning

For each of them, an adversary simply needs to be in the same local network as the victim such that they share a cache. None of the attacks actually involve illegally sniffing on other people's traffic. Instead, attackers can simply request items from the caches like any legitimate user.

5.2.1 Cache Enumeration

There are mainly two reasons why someone might want to enumerate a cache. If the cache population, i.e. the people who share a cache, is small, it could be possible to infer who has requested specific content just by looking at all cached content. In many cases, the name of the content will involve some user-specific part, e.g. `/provider/mail/username`, that immediately gives away the identity of the requesting person. In addition to that, the signature of the content source could be enough to infer the communication partner. For censors, it might even be sufficient to find out that *someone* queried specific content. Otherwise, finding out which content is in the cache is a first step before a Timing Attack or cloning a conversation.
The feasibility of Cache Enumeration arises from routers allowing users to either directly request all cached data, e.g. with a command like `ccnls` in CCNx, or by using a combination of prefix matching and the exclude functionality of NDN as explained by Lauinger et al. [41]. Prefix matching is performed on every query, so if the cache includes `/email/work/2015` and `/email/private/2015` and the adversary requests `/`, an arbitrary cached content is returned because all entries match. Then, the exclude functionality is used to query again excluding the first content name. While this may take lots of requests for a large cache, the distance to the cache is very small. Still, it is unlikely that the attacker will get a consistent view of the cache.

5.2.2 Timing Attack

Timing Attacks aim at finding out if and when a user requested a specific piece of content. The adversary can measure the time it takes for the content to arrive and compare it to previously measured times to find out whether his request resulted in a cache hit or not. Therefore, measuring round-trip-times is a preliminary step for a Timing Attack. It also requires the attacker to know in advance which content the user might query. Censors, for example, could have a number of blacklisted files and check whether a user requested them. But even local adversaries with limited resources can execute such an attack to spy on people within the same cache population. Lauinger et al. [41] describe how Timing Attacks can be carried out invasively and non-invasively. Here, *invasive* means that the cache treats a request as any other request and possibly updates its state, while a non-invasive request would not influence the cache's state and simply retrieve the cache's content. Which method is chosen depends solely on the cache itself and whether it allows non-invasive queries. Disallowing them makes it harder for the adversary, although not impossible. In the invasive scenario, the attack depends on the cache replacement policy. For example in the case of FIFO (First In First Out), LRU (Least Recently Used) or Random Replacement, the attacker first needs to find out the *cache lifetime* or *characteristic time* t_c of the cache, i.e. the expected time for which the cache stores a piece of content. For LRU, it is the time a file remains in the cache while not being queried. The attacker can then repeatedly request the content every $t_c + \epsilon$ time units for a small ϵ. If he gets a cache hit, someone within the cache population queried the content within the last interval. Especially if a user-specific content name is used, the attacker knows who requested the content. Since the characteristic time is determined as an expected value, the attacker will get a result that is true only with a certain confidence. Another problem for the adversary is that the targeted user could request the content shortly after the attacker, i.e. within ϵ time units, and the request would therefore not be detected. CCN offers a way to overcome this, though: Every cached content is stored as chunks of 4KB which can individually be requested. The attackers can thus iterate over the chunks of the given content and query one of them every t_c time units. Lauinger et al. [41] call this Parallel Cache Probing. It is important to notice that these methods are dependent on the cache replacement strategy. With a more complex method using randomness it could become infeasible for a local adversary to launch a Timing Attack. For example, in order to understand which content is cached, one might have to observe all requests at a router. This could restrict this attack to more powerful adversaries like a censoring country which might have access to the router anyway and will not use this method.

5.2.3 Conversation Cloning

The idea behind Conversation Cloning is that an attacker tries to get the whole content corresponding to a data flow. This flow could, for example, be a video call using Voice-over-CCN (VoCCN) [36]. The content itself is assumed to be encrypted. But by reassembling the messages of the conversation, the adversary could potentially find a side channel and infer privacy-related information about the content. Possible side channels include the size of the messages and their timings. A similar attack is possible on VoIP [54], so it is likely to be a threat for VoCCN, too.
Somehow, adversaries need to find out about the names that are used in the data flow. It is likely to have some kind of serial number, e.g. `/voccn/call/alice/1`, followed by `/voccn/call/alice/2` and so on. With the previously discussed Cache Enumeration, the attacker can easily find out about a current sequence number and possibly the naming scheme. This enables him to predict names of future messages and to request and receive the messages in real-time like the legitimate receiver. As discussed in the next section, the naming issue can be overcome by encrypting parts of the names or even the whole name. If the attacker is not able to retrieve the content in the right order, he is at a significant disadvantage, because he can not infer the order from the content itself as it is encrypted. The main problem is still that Cache Enumeration is possible in the first place. Using the exclude functionality, every message with the known routing prefix `/voccn/call/alice/` can be

queried by attacker.

5.2.4 Countermeasures

This section contains a list of possible countermeasures against the previously described attacks. They include - but are not limited to - the work of Lauinger [40], Lauinger et al. [42, 41], Chaabane et al. [17], Acs et al. [7]. Most of them affect the efficiency. As a result, we get a trade-off between privacy and efficiency.

1. Disallow caching in general. This eliminates one of the main benefits of Content Centric Networks.
2. Disallow non-invasive queries, i.e. a query should always affect the cache's state and not just allow users to enumerate the contents without changing the state.
3. Disallow techniques that are used for Cache Enumeration, such as `ccnls`. In the context of DNS Snooping such requests are often called non-recursive.
4. Restrict or completely disallow the exclude functionality of caches.
5. Disallow requesting chunks of a piece of content to prevent Parallel Cache Probing.
6. Encrypt names in interactive conversations. Encryption prevents the adversary from predicting names. The content is not meant for people who do not know about the naming scheme anyway. Therefore, the efficiency of the network is not degraded.
7. Tunnel traffic using an anonymity system. This has basically the same effect on Cache Privacy as Encryption. AND$\bar{a}$NA [26] is an example of such an anonymity system.
8. Use a more complex cache replacement policy. The Timing Attacks rely on knowledge about the cache lifetime. If the cache uses a complex strategy, the attacker has a harder time understanding it. Attacks based on the cache lifetime might not be possible at all any more.
9. Only cache popular content. Popular content is requested often and by multiple people. This can be enforced with a cache replacement policy that adapts to the content that is forwarded by the router. Current research is focusing on such topics.
10. Make cache lifetimes very short. This limits the time for adversaries to retrieve the requested content, but at the same time affects the performance.
11. Choose cache lifetime randomly for each cached file to prevent attackers from measuring the cache's characteristic time.
12. Increase the anonymity set, i.e. the cache population. This results in more people being served by the same cache and thus in reduced efficiency.
13. Add a minimum response delay in case of a cache hit. The actual delay could be chosen randomly in order to prevent attackers from measuring the minimum response delay. It obviously increases the total delay, but if chosen carefully the adversary can not distinguish whether the content was cached in the closest cache or several hops away.
14. Do not cache private content. This is similar to only caching popular content, but instead identifying content that should not be cached. The key observation is that private content should only ever be received by the communicating parties and caching it will neither improve their privacy nor the overall performance of the network, because nobody else can possibly legitimately want the content. Based on this, Lauinger et al. [42, 41] propose selective countermeasures, i.e. countermeasures that are only applied to privacy-related content. One way to implement this is by using a do-not-cache flag in interests. But privacy-aware users might decide to use it on all their interests, so it might be better to have the content origin set the do-not-cache flag. The disadvantage of this option is that the receiver has no choice. Furthermore, caches of routers on the return path could simply decide to ignore the flag and still cache the content.

Apart from these countermeasures, it might already be enough to detect attacks on Cache Privacy. If an attacker is identified, his requests could be ignored partially or completely to resolve the situation. The detection is unfortunately not easy and prone to finding false positives. The detection methods include the following:

1. Edge routers keep track track of how often the same content is queried from a single interface. If periodic querying occurs, it is assumed to be a Timing Attack or Cache Enumeration.
2. Edge routers remember the recent hit rate for each interface. If an unusually high hit rate is detected, it is likely because of Cache Enumeration.
3. Edge routers keep track of requests using the exclude functionality. As a consequence, they can find out if somebody is trying to clone the conversation.

All these methods should be deployed as close to the users as possible. Still, all of them are suboptimal. Firstly, periodic querying might be necessary in a company that wants to know the state of some content. For example, a news agency always has to know whether it is missing a story that another agency has already put on their website. Secondly, a high hit rate could be accidental. For example, if multiple people are watching the same movie, but they did not start exactly at the same time, there will be lots of cache hits resulting in false positives. Thirdly, the exclude functionality can be used in a legitimate way. If a user makes extensive use of it, he could falsely be detected as an attacker.

5.3 Name Privacy

With a content-oriented architecture, the performance benefits come from having a publicly known and often human-readable name for the content and thus being able to match

requested content names to cached content and to aggregate interests. But users sacrifice their privacy for using these names. Assume that a user lives in a censoring country and wants to look at blocked content. Retrieving the content with its human-readable name is impossible, because the censors can filter out unwanted names. If the content were already cached at the time the censors decide to block it, they could simply delete content with the given name from all caches. A powerful adversary could even store interests to analyze them later. Caches allow censors to find unwanted content even after it is sent. Even for privacy-aware users that do not want anybody to track their interests it is not desirable to send names in the clear. As a result, Name Privacy is a challenge in Content Centric Networking.
There are conceivable countermeasures, some of which are inspired by techniques used in the current Internet.

1. Encrypt content names. If only the last part of the name is encrypted, e.g. `/provider'/mail'/username'/a2e13f7b5`, there is still a fairly good chance to guess what the content is. Encrypting the whole name would result in routing problems because the hierarchical structure is not used any more. Instead, a part of the name should be encrypted, e.g. `/provider'/mail'/c298a67fe'/a2e13f7b5`. The encryption can not be extended over the name of the provider, because the interest would not arrive there due to a non-matching name prefix. In many cases, though, censors will completely block a provider. We conclude that the hierarchy of names helps censors distinguish between benign and unwanted content. As a consequence, encryption of names is only a viable option for increased privacy, but not sufficient for censorship circumvention. It is worth noting that the encryption has to be negotiated before the interest and that advantages of Content Centric Networks such as caching are lost.
2. Decoy Routing. Decoy Routing is essentially redirecting traffic once it is out of the reach of a censor. As effective as it is, Decoy Routing would require architectural changes. For a comprehensive description and analysis, we refer to [38, 35].
3. Change names. To circumvent censors' blacklist of names, one could simply change the name in order to make it seem to the censor as if the content is normal and should not be filtered out. The problem is that the censors are likely to get suspicious if a content provider that is known for producing unwanted content suddenly offers lots of unblocked content. To check the assumption, they can query the content themselves. Furthermore, suspicious content providers are likely to be completely blocked by censors anyway. Therefore, changing names is at least not a permanent solution to censorship circumvention.
4. Ephemeral names. The idea is to generate new names for blocked content. Users should be able to predict the names while censors should not. This could maybe be achieved by having a separate generation algorithm for every user. In any case, with ephemeral names the benefits of the content-oriented architecture are lost.
5. Anonymity Systems. Relaying encrypted traffic through multiple other nodes is certainly an option, although again the advantages of Content Centric Networking are not used. An anonymity system similar to Tor in the current Internet is AND$\bar{a}$NA. We describe AND$\bar{a}$NA in Section 5.9

All in all, there is no perfect solution so far. This will surely be addressed by researchers.

5.4 Signature Privacy

While we previously stated that integrity verification is a benefit of Content Centric Networking, it also provides problems. Content providers are identifiable by their signature, so censors can block content if they recognize an unwanted publisher. To mitigate this risk, we suggest two countermeasures.

1. *Group* or *ring signatures.* A number of content providers share a group signature. This can be administrated by a group manager (group signatures) or through interaction of the group members (ring signature). If a publisher whose content is filtered out by censors uses a group signature that he shares with a number of unblocked and popular publishers, there is a chance the censors will not block his content any more. Otherwise, the censors risk losing the services of the other popular publishers as well. It is unclear, though, whether popular services would take the risk of getting blocked themselves without incentives. There has been lots of research on group signatures that we recommend for more information, specifically [18, 15, 13, 16].
2. Ephemeral identities. A content provider could generate ephemeral identities for every piece of content. With a signature that is unknown to the censor, the content could go unblocked. Even if the censor decides to check the origin of the content, it would at least affect the performance of the censors network and add expensive computations. Whether or not censors are willing to do this remains to be seen.
3. Proxies. Instead of ephemeral identities, proxies could be used. The signature then becomes the signature of the proxy. If a proxy is used for unwanted content frequently, censor will eventually block the proxy's forwarded content.

All the proposed solutions seem promising. The effects are all speculative, though, until they are actually used in a real-world scenario.

5.5 Content Privacy / Access Control

As in the current Internet, access control is enforced by content providers through encryption. This becomes even more important since their content might not even be coming directly from the publisher, but from a cache. If the encryption is specific to one user, the effect of caching is nullified. Therefore, Fiat et al. [29] propose broadcast encryption which allows n users with different keys to decrypt the same encrypted content. While this allows effective caching, this method produces keys of length $\mathcal{O}(\sqrt{n})$ and increases the length of the encrypted content by a factor $\mathcal{O}(\sqrt{n})$. A

different solution is atomic proxy cryptography offered by Blaze et al [14]. The content provider encrypts content such that the first user can decrypt it and provides a proxy with a re-encryption key. The proxy has no knowledge about the content and simply applies the key and delivers the re-encrypted content to the second user, re-encrypts for the third user etc. The main problem of atomic proxy cryptography is the use of asymmetric keys that require computationally expensive operations.

5.6 Accountability

The previously claimed advantage of having no addresses can be a problem. Law enforcement, for example, has to be able to trace back who requested illegal content. There are basically two outcomes: Either law enforcement is given access to the whole infrastructure such that they can keep track of interests in illegal content, or not. If not, users can not be held accountable for requesting illegal content. This aspect of Content Centric Networking has barely been studied, but is of great importance from a legal perspective.

5.7 Revocation and Removal

There are many reasons why somebody would like content to be remove content from caches:

- Revocation. Content can become outdated, especially if hosts repeatedly request the same piece of content and it has been in the cache for a long time. Content providers need a mechanism to either update or at least revoke outdated content.
- *Content Poisoning.* Fake content is in the cache and should be removed. Ghali et al. [32] examine this in detail. Basically, content poisoning occurs when the signature is invalid or not verifiable. An adversary can try to distribute the fake copy of content. This problem can easily be mitigated by having routers verify signatures, but - as discussed earlier - this causes the routers to perform expensive computations.
- *Cache Pollution.* This attack is discussed in Section 5.1. An attacker tries to fill the cache with arbitrary data to degrade the network's performance.
- Illegal content removal. Criminals might try to keep illegal data in a cache by repeatedly querying it. The availability of illegal data within the network infrastructure is not acceptable, so there should be a removal mechanism.

While there are more sophisticated countermeasures, we restrict the following list to general purpose measures that achieve the content removal:

1. Explicit Removal. The service provider owning the cache could manually access and remove content. Assuming a large number of such cases, this method is infeasible.
2. Blacklist broadcasting. If the detection of unwanted content is automated, a blacklist of content names could be created that is broadcasted among routers. If a router receives the list, it deletes all cached content that is present in the blacklist. The broadcast messages cause a large communication overhead.
3. Periodic revalidation. Routers periodically verify that the cached content is still up-to-date. Like the previous approaches, periodic revalidation imposes a communication overhead for the routers.

There is certainly more research to be done on content removal from caches, because the described methods all have issues. Previous research has mainly aimed at detecting unwanted content before it is stored at all, but there need to be mechanisms to remove content that is not detected or identified as illegal later.

5.8 Privacy Enhancing Technologies for Content Centric Networking

There are many *Privacy Enhancing Technologies* (PET) for current Internet, but it has yet to be checked which are applicable to Content Centric Networking. Also, there might be new techniques conceivable. In this section, we present AND$\bar{a}$NA, which is to our knowledge the only currently existing anonymizing system for NDN.

5.9 ANDaNA

AND$\bar{a}$NA [26] is an attempt to build Tor [28] for NDN. It is designed as an overlay network on top of the NDN architecture. Important elements of Tor were used, such as *Onion Routing*. The goal of AND$\bar{a}$NA is to provide anonymity through layered encryption. The assumption is that there is no adversary with the power of global surveillance. Under that assumption, traffic is relayed twice through so-called *anonymizing routers* (AR) which are hosts or routers distributed all over the world. Each AR adds another layer of encryption. Compared to Tor, only two relays are necessary to achieve the same degree of anonymity. This is due to the absence of addresses in Content Centric Networking. The first step in a communication through AND$\bar{a}$NA is building an ephemeral circuit. It is used for only one interest and closed afterwards. For that, the user distributes a separate symmetric key to each of the ARs which are chosen from a public list in advance. Once the ephemeral circuit is set up, the interest is sent through the tunnels. The last part from the exit AR to the content provider is the actual communication as it is normally performed. After receiving the content, the exit router encrypts the content, original name and signature with the symmetric key. This ciphertext is the content that he forwards to the entry AR. The content and name are signed by the exit AR. The same happens at the entry AR. When the user finally receives the content, he has to decrypt it with the symmetric keys and verify the producer's signature.
The crucial point about anonymizing systems is that their delays should be relatively small in order to still provide a usable service. In their evaluation, DiBenedetto et al. [26] show that AND$\bar{a}$NA outperforms Tor in the current Internet for small files up to 10MB. Furthermore, the caching benefits of Content Centric Networks are lost because of the encryption. AND$\bar{a}$NA does not provide solutions for most known

problems of Tor such as blocked relays and the distribution of a list of relays.

6. RELATED WORK

Chaabane et al. [17] thoroughly describe challenges to privacy in Content Centric Networks. They divide privacy threats into Cache, Content, Name and Signature Privacy and discuss possible countermeasures. Their work provides a great overview and starting point for further research on the topic.

Lauinger [40] looks mainly at DoS- and Cache Privacy-related problems of Content Centric Networking. His detailed analysis not only shows the weaknesses of the approach, but he also provides a comprehensive discussion of potential solutions.

Acs et al. [7] examine Cache Privacy for NDN in great detail. They show how easy adversaries can find out about previously queried content and thus reconstruct a user's communication. In order to overcome this issue, they propose several countermeasures like delays and explicit privacy bits. Similarly, Lauinger et al. [41] look at the effect of different caching strategies on Cache Privacy. They also describe how caches can enumerated by simply using functionality provided by CCN.

Arianfar et al. [10] consider Content Centric Networks as a step backwards in terms of privacy. To counter this, they suggest mixing a censored file with a cover file to achieve computational asymmetry. While this method does not provide guarantees, it increases the effort censors have to invest.

DiBenedetto et al. [26] propose AND$\bar{a}$NA, an onion routing network designed as an overlay network to NDN. Similarly to Tor, AND$\bar{a}$NA aims at providing anoymity to users by relaying traffic and with layered encryption.

Ghali et al. [31] discuss content poisoning extensively. In order to prevent this problem, they argue that content verification should be left to the communicating parties instead of the routers. They propose small changes in the communications of NDN in order to establish trust management on the network layer.

Xie et al. [55] propose CacheShield, a robust cache scheme applicable to CCN. CacheShield tries to optimize performance by caching only popular content. Their studies show that CacheShield also helps to mitigate cache pollution attacks. Teoli [49] evaluated CacheShield more extensively and confirmed its effectiveness. Conti et al. [24] argue that CacheShield requires too much storage space. As an alternative, they propose a lightweight mechanism for the detection of cache pollution.

Ghali et al. [31] describe content poisoning in detail and come up with a solution that involves trust management on the network layer. Effectively, the introduce a so-called Interest-Key-Binding (IKB) which happens when sending the interest.

A lot of research groups have focused on the detection and mitigation of Denial of Service attacks in Content Centric Networks or Interest Flooding Attacks (IFA). Gasti et al. [30] broadly discuss DoS and DDoS attacks in NDN and suggest a number of mitigation techniques. Compagno et al. [22] experimented with a countermeasure based on router statistics. They show that even under attack, they can achieve a forwarding percentage of more than 80% of legitimate traffic. Ding et al. [27] detect IFAs by using an entropy-based model. Dai et al. [25] suggest using tracebacks in case of a suspected attack. By using the PITs of routers, they manage to find the origins of the attack and limit their access to the network by dropping interests. Wang et al. [52] describe how attackers could circumvent interest aggregation by requesting non-existent content. As a countermeasure, they suggest a threshold-based detection and mitigation scheme (TDM) that recognizes when too many interests time out. As an alternative, Wang et al. [53] propose Disabling PIT Exhaustion (DPE) to reduce the effect of IFAs by directly recognizing malicious requests before the PIT is full. Afanasyev et al.[8] performed large-scale simulations to test the effectiveness of several mitigation algorithms. Especially the satisfaction-based pushback algorithm, which enforces a limit on the number of forwarded interests based on an interface's interest satisfaction ratio, achieved promising results. Li et al. [43] propose Interest Cash, a countermeasure against IFA that mandates solving a puzzle before being allowed to send interests. Their evaluations show that the an attacker needs more than 300 times more resources than a publisher to launch a successful IFA against him.

7. CONCLUSION

In this paper, we provide an overview of how various aspects of security and privacy change with Content Centric Networks. Even though security is a main design principle, the architecture allows for new attacks such as Interest Flooding Attacks, Cache Pollution and Content Poisoning. Furthermore, privacy is not taken into account which leads to new problems categorized as Cache, Content, Name and Signature Privacy. Overall, Content Centric Networking is intended to be a scalable and efficient architecture. It is neither completely beneficial nor completely detrimental for security and privacy. Instead, some aspects were improved while others offer new vulnerabilities. In general, we conclude that there is a tradeoff between the effectiveness of caches and privacy. This implies a change of methods for both attackers and defenders. In the current Internet, security was mainly built on top of the architecture. It remains to be seen whether more security and privacy can be added to Content Centric Networks. Considering that actual projects like CCNx and NDN are still being improved and changed, it is vital for researchers to find flaws and solutions in the architecture now while changes are still possible.

8. ACKNOWLEDGMENTS

Thanks to Amir Houmansadr for stimulating discussions in several meetings throughout the semester.

9. REFERENCES

[1] CCNx homepage. `http://www.ccnx.org/`. Accessed: 2015-02-22.

[2] Mobilityfirst homepage. `mobilityfirst.winlab.rutgers.edu`. Accessed: 2015-04-17.

[3] Named data networking homepage. `named-data.net`. Accessed: 2015-02-20.

[4] NSF future internet architecture project homepage. `www.nets-fia.net`. Accessed: 2015-04-19.

[5] Xia homepage. `mobilityfirst.winlab.rutgers.edu`. Accessed: 2015-04-16.

[6] M. Aamir and S. M. A. Zaidi. Denial-of-service in content centric (named data) networking: a tutorial and state-of-the-art survey. *Security and Communication Networks*, 2014.

[7] G. Acs, M. Conti, P. Gasti, C. Ghali, and G. Tsudik. Cache privacy in named-data networking. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pages 41–51. IEEE, 2013.

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

[9] A. Anand, A. Gupta, A. Akella, S. Seshan, and S. Shenker. Packet caches on routers: the implications of universal redundant traffic elimination. In *ACM SIGCOMM Computer Communication Review*, volume 38, pages 219–230. ACM, 2008.

[10] S. Arianfar, T. Koponen, B. Raghavan, and S. Shenker. On preserving privacy in content-oriented networks. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pages 19–24. ACM, 2011.

[11] S. Arianfar, P. Nikander, and J. Ott. On content-centric router design and implications. In *Proceedings of the Re-Architecting the Internet Workshop*, page 5. ACM, 2010.

[12] S. Arianfar, P. Nikander, and J. Ott. Packet-level caching for information-centric networking. In *ACM SIGCOMM, ReArch Workshop*, 2010.

[13] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Advances in Cryptology-Eurocrypt 2003*, pages 614–629. Springer, 2003.

[14] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology-EUROCRYPT 1998*, pages 127–144. Springer, 1998.

[15] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Advances in Cryptology–CRYPTO 2004*, pages 41–55. Springer, 2004.

[16] J. Camenisch. Efficient and generalized group signatures. In *Advances in Cryptology-EUROCRYPT 1997*, pages 465–479. Springer, 1997.

[17] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun. Privacy in content-oriented networking: Threats and countermeasures. *ACM SIGCOMM Computer Communication Review*, 43(3):25–33, 2013.

[18] D. Chaum and E. Van Heyst. Group signatures. In *Advances in Cryptology Eurocrypt 1991*, pages 257–265. Springer, 1991.

[19] S. Choi, K. Kim, S. Kim, and B.-h. Roh. Threat of dos by interest flooding attack in content-centric networking. In *Information Networking (ICOIN), 2013 International Conference on*, pages 315–319. IEEE, 2013.

[20] D. Clark. The design philosophy of the darpa internet protocols. *ACM SIGCOMM Computer Communication Review*, 18(4):106–114, 1988.

[21] D. Clark, R. Braden, K. Sollins, J. Wroclawski, and D. Katabi. New arch: Future generation internet architecture. Technical report, DTIC Document, 2004.

[22] A. Compagno, M. Conti, P. Gasti, and G. Tsudik. Ndn interest flooding attacks and countermeasures. In *Annual Computer Security Applications Conference*, 2012.

[23] M. Conti, S. Chong, S. Fdida, W. Jia, H. Karl, Y.-D. Lin, P. Mähönen, M. Maier, R. Molva, S. Uhlig, et al. Research challenges towards the future internet. *Computer Communications*, 34(18):2115–2134, 2011.

[24] M. Conti, P. Gasti, and M. Teoli. A lightweight mechanism for detection of cache pollution attacks in named data networking. *Computer Networks*, 57(16):3178–3191, 2013.

[25] H. Dai, Y. Wang, J. Fan, and B. Liu. Mitigate ddos attacks in ndn by interest traceback. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 381–386. IEEE, 2013.

[26] S. DiBenedetto, P. Gasti, G. Tsudik, and E. Uzun. Andana: Anonymous named data networking application. *arXiv preprint arXiv:1112.2205*, 2011.

[27] K. Ding, Y. Liu, H.-H. Cho, H.-C. Chao, and T. K. Shih. Cooperative detection and protection for interest flooding attacks in named data networking. *International Journal of Communication Systems*, 2014.

[28] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. Technical report, DTIC Document, 2004.

[29] A. Fiat and M. Naor. Broadcast encryption. In *Advances in Cryptology-CRYPTO 1993*, pages 480–491. Springer, 1994.

[30] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. DoS and DDoS in named data networking. In *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, pages 1–7. IEEE, 2013.

[31] C. Ghali, G. Tsudik, and E. Uzun. Elements of trust in named-data networking. *arXiv preprint arXiv:1402.3332*, 2014.

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

[33] A. Ghodsi, S. Shenker, T. Koponen, A. Singla, B. Raghavan, and J. Wilcox. Information-centric networking: seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, page 1. ACM, 2011.

[34] B. Hamdane, A. Serhrouchni, A. Fadlallah, and S. Fatmi. Named-data security scheme for named data networking. In *Network of the Future (NOF), 2012 Third International Conference on the*, pages 1–6. IEEE, 2012.

[35] A. Houmansadr, E. L. Wong, and V. Shmatikov. No direction home: The true cost of routing around decoys. In *Proceedings of the 2014 Network and Distributed System Security (NDSS) Symposium*, 2014.

[36] V. Jacobson, D. K. Smetters, N. H. Briggs, M. F.

Plass, P. Stewart, J. D. Thornton, and R. L. Braynard. Voccn: voice-over content-centric networks. In *Proceedings of the 2009 workshop on Re-architecting the internet*, pages 1–6. ACM, 2009.

[37] A. Karami and M. Guerrero-Zapata. A hybrid multiobjective rbf-pso method for mitigating dos attacks in named data networking. *Neurocomputing*, 151:1262–1282, 2015.

[38] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer. Decoy routing: Toward unblockable internet communication. In *USENIX workshop on free and open communications on the Internet*, 2011.

[39] J. Kurose. Information-centric networking: The evolution from circuits to packets to content. *Computer Networks*, 66:112–120, 2014.

[40] T. Lauinger. *Security & scalability of content-centric networking*. PhD thesis, TU Darmstadt, 2010.

[41] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda. Privacy implications of ubiquitous caching in named data networking architectures. Technical report, Technical report, TR-iSecLab-0812-001, iSecLab, 2012.

[42] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda. Privacy risks in named data networking: what is the cost of performance? *ACM SIGCOMM Computer Communication Review*, 42(5):54–57, 2012.

[43] Z. Li and J. Bi. Interest cash: an application-based countermeasure against interest flooding for dynamic content in named data networking. In *Proceedings of The Ninth International Conference on Future Internet Technologies*, page 2. ACM, 2014.

[44] D. Naylor, M. K. Mukerjee, P. Agyapong, R. Grandl, R. Kang, M. Machado, S. Brown, C. Doucette, H.-C. Hsiao, D. Han, et al. Xia: architecting a more trustworthy and evolvable internet. *ACM SIGCOMM Computer Communication Review*, 44(3):50–57, 2014.

[45] M. Paknezhad and M. Keshtgary. Security and privacy issues of implementing cloud computing on ndn. 2014.

[46] J. Pan, S. Paul, and R. Jain. A survey of the research on future internet architectures. *Communications Magazine, IEEE*, 49(7):26–36, 2011.

[47] J. Rexford and C. Dovrolis. Future internet architecture: clean-slate versus evolutionary research. *Communications of the ACM*, 53(9):36–40, 2010.

[48] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri. Mobilityfirst future internet architecture project. In *Proceedings of the 7th Asian Internet Engineering Conference*, pages 1–3. ACM, 2011.

[49] M. Teoli. *cache pollution attacks and detection in named data networking*. PhD thesis, University of Padua, 2013.

[50] A. Venkataramani, J. F. Kurose, D. Raychaudhuri, K. Nagaraja, M. Mao, and S. Banerjee. Mobilityfirst: a mobility-centric and trustworthy internet architecture. *ACM SIGCOMM Computer Communication Review*, 44(3):74–80, 2014.

[51] M. Wählisch, T. C. Schmidt, and M. Vahlenkamp. Backscatter from the data plane–threats to stability and security in information-centric network infrastructure. *Computer Networks*, 57(16):3192–3206, 2013.

[52] K. Wang, H. Zhou, H. Luo, J. Guan, Y. Qin, and H. Zhang. Detecting and mitigating interest flooding attacks in content-centric network. *Security and Communication Networks*, 7(4):685–699, 2014.

[53] K. Wang, H. Zhou, Y. Qin, J. Chen, and H. Zhang. Decoupling malicious interests from pending interest table to mitigate interest flooding attacks. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 963–968. IEEE, 2013.

[54] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted voip conversations. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 35–49. IEEE, 2008.

[55] M. Xie, I. Widjaja, and H. Wang. Enhancing cache robustness for content-centric networking. In *INFOCOM, 2012 Proceedings IEEE*, pages 2426–2434. IEEE, 2012.

[56] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos. A survey of information-centric networking research. *Communications Surveys & Tutorials, IEEE*, 16(2):1024–1049, 2014.

[57] R. You, R. Luo, and X. Lai. Detecting and mitigating interest flooding attack in content centric networking. *Advances in Computer Science and Technology*, 65:251, 2014.

[58] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al. Named data networking (ndn) project. *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.