# On the Possibility of Mitigating Content Pollution in Content-Centric Networking

Igor Ribeiro, Antonio Rocha, Célio Albuquerque
Computing Institute
Fluminense Federal University
Niterói, Brazil
(iribeiro,arocha,celio)@ic.uff.br

Flávio Guimarães
Cyber Defense Division
Brazilian Navy
Rio de Janeiro, Brazil
queiroz@dctim.mar.mil.br

***Abstract*—Content-Centric Networking is an architecture proposal for the future Internet that brings fundamental changes in the way the network operates. Contents are identified and requested based on their names and for security reasons they must be digitally signed by their publishers. Even though this new architecture was designed to be safe, one potential security threat is that malicious publishers may create polluted versions of legitimate contents, reducing their availability and degrading network resources. Because of the non-negligible overhead of checking a large number of signatures, it is not feasible to make it a mandatory task for every router, especially in the network core. In this paper, we propose CCNCheck: a mechanism in which CCN routers probabilistically check the content signatures. We evaluate the mechanism against simulations and found evidences that using CCNCheck increases the fraction of recovered contents and decreases the wastage of network resources.**

## I. Introduction

Content-Centric Networking (CCN) [5] is a proposal for the future Internet that moves the focus of the network from the content's location to the content itself. In order to request a content, a user simply creates an interest packet including the name of the desired content. The requested data can be retrieved from any node storing it and is sent back to the user within a data packet. To improve content availability and retrieval efficiency, network routers cache data packets before forwarding them. Consequently, future interests for a previously requested content can be immediately replied by a router storing a correspondent cached data packet. In CCN jargon, users that request contents are called Consumers and those that provide contents are called Publishers.

As no source or destination information is provided in CCN packets, consumers can not precisely determine from which repository a received content was retrieved. The consequence is that the authenticity and integrity of the contents are an issue. CCN solves this problem by enforcing that publishers digitally sign every content they provide. Further, to improve security, the publishers actually sign the association between the content and its name. In order to check the authenticity of a content, the consumer must get the publisher's public key, which can be included in the data packet in the form of a Public Key Infrastructure (PKI) certificate, for example. If the signature verification fails, the consumer must discard the content [11]. More information on CCN and other content-oriented architectures can be found in [4].

### A. The Content Pollution Problem

Due to its content-oriented paradigm, the main objective of CCN architecture is to allow users to retrieve contents in a straightforward and efficient way. Malicious users may take advantage of the high availability provided by CCN to disseminate malicious contents in order to cause harm to legitimate users or to the network infrastructure. In the context of the present work, we define the following types of content pollution attacks against CCN:

**Content Renaming:** the name of a given content is replaced by a fake one;
**Content Corruption:** the content itself or its metadata is altered in such a way that it becomes useless to the users; and
**Content Falsification:** the publisher's public key informed on the data packet does not pair with the private key used on the signature process. Even if signature verification succeeds, the informed publisher's public key may belong to an untrustworthy real world entity. In this case, the detection of the attack depends on the trust management mechanism used.

When a polluted content is forwarded to the consumer, it is cached in the routers along the path. Upon the receipt of the polluted content the consumer verifies its signature and discards it. If the consumer still wants to retrieve a legitimate copy of the desired content, she may request it again. However, since its border router already has a polluted copy in cache, it will be sent back to the consumer. Again, the consumer will check the signature and will discard the content. Depending on the pollution level of the network, users may be prevented from retrieving legitimate contents, thus resulting in a DoS attack. In addition, processing and forwarding polluted contents waste network resources and reduce its efficiency.

One possible solution for this problem would be to bring the signature verification to the core of the network. Unfortunately, due to the processing overhead imposed by signature verification, this approach is infeasible [7].

978-1-4799-3780-6/14/$31.00 ©2014 IEEE

### B. Related Works

Since CCN is a new network architecture not yet implemented in large scale, much of its most relevant security issues are still being discussed. Specifically, CCN has been found to be vulnerable to Denial of Service (DoS) attacks [3], [7] and prone to privacy violation attacks [8], [9], [10]. In addition, as CCN is essentially a network of caches, it is also vulnerable to cache pollution attacks, such as false locality and locality disruption attacks [12].

Besides these research fields, CCN's vulnerability to Content Pollution attacks has been receiving little attention by the research community. Actually, to the best of our knowledge, the only other work that addresses this subject is performed by *Ghasti et al.* [7]. The authors proposed one approache to mitigate the problem, that requires routers to verify the signature of a random portion of all cached contents. It is important to note that the signature verification is performed only to the contents stored in the router's cache. On the other hand, the mechanism proposed in the present work aims to make routers perform the verification as part of the forwarding decision. Consequently, `CCNCheck` can be understood as a complement to the approache proposed by *Ghasti et al.*.

## II. CCNCheck

The main objective of `CCNCheck` is to increase the availability of legitimate contents and to reduce the wastage of network resources due to the processing and forwarding of polluted contents. Our approach does bring the signature verification process to the core of the network. However, instead of checking the signature of all contents traversing the network, `CCNCheck` verifies the signature of only a random subset of these contents.

It is not difficult to foresee that the verification probability defined in the mechanism plays a key role in the reduction of polluted contents in the network. However, it is also clear that the greater this probability value is, the higher is the overhead imposed to the network routers. Thus, there is a trade off in this value and it must be chosen carefully to ensure a good balance between the overhead imposed by the signature check overhead and the efficiency in reducing the dissemination of polluted contents.

### A. Design

Originally, CCN allows that decisions regarding the forwarding of interest packets are taken hop by hop, based on the forwarding policy employed. On the other hand, data packets must always follow the reverse path traversed by the corresponding interest. With the implementation of `CCNCheck`, this behavior changed, introducing the concept of Data Packet Forwarding Policy. With `CCNCheck`, the signature of data packets are probabilistically checked. Consequently, legitimate contents are always cached and forwarded while the polluted ones are immediately dropped.

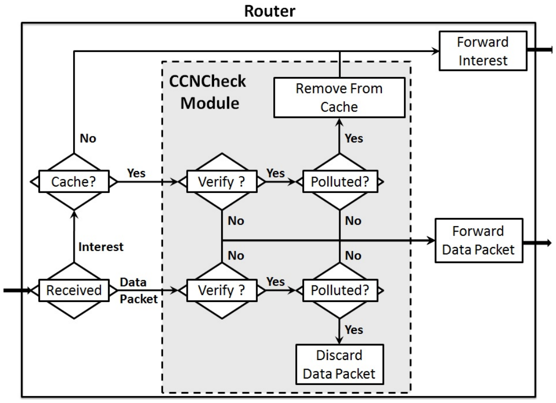

Fig. 1. Packets flowing in a ccn-based router with `CCNCheck` module

Figure 1 shows a flow diagram representing the steps followed by a router when a packet is received. If the received packet is an interest packet, then the router needs to check whether it has the requested content in its cache. In case it does, before the router replies the received interest, it sends the cached content to the `CCNCheck` module. The content's signature will be checked based on a given probability. If the content happens to be polluted, then it will be removed from the router's cache and the received interest is forwarded according to the forwarding policy in use. On the other hand, if the content is legitimate, then it will reply the received interest.

If the received packet is a data packet, it must be forwarded through the reverse path given by the corresponding interest. However, before forwarding the data packet, it is sent to the `CCNCheck` module. Again, the content's signature is verified based on a given probability. If the content is polluted, then the data packet is simply dropped. On the other hand, if the content is legitimate, it is cached by the router and forwarded as usual.

### B. Deployment Approaches

`CCNCheck` is flexible enough to allow the definition of specific strategies for different scenarios. For example, it is possible to configure all routers in a network to use the same static verification probability. In addition, one can choose to adjust the verification probability dynamically in order to adapt `CCNCheck` to the current pollution level.

In the present work, we analyse the improvements on the pollution mitigation caused by the use of `CCNCheck` when all routers check content signatures based on the same static probability.

## III. Simulations

To evaluate the efficiency of the proposed mechanism, we conducted a set of simulation experiments using the NS3 simulator [1] configured with the NDNSim module [2], which implements the CCN stack. Table I summarizes the simulation parameters.

### A. Network Topologies

**Grid:**The grid topology has dimensions 21x21 totaling 441 nodes and 840 links between them. The malicious publisher occupies the position (1,10), the legitimate

TABLE I
SIMULATION PARAMETERS FOR GRID AND ROCKETFUEL TOPOLOGIES

| Parameter | Grid | Rocketfuel |
|---|---|---|
| Data packet size (bytes) | 1024 | 1024 |
| Number of requests (contents) | 20 | 20 |
| Consumer request rate (interests/sec.) | 10 | 10 |
| Request timeout | Dynamic | Dynamic |
| Maximum retries per content | 10 | 10 |
| Forwarding strategy | Flooding | Flooding |
| Simul. runs per prob. value (p) | 500 | 25000 |

publisher occupies the position (1,12) and the consumer is positioned at (21,11). All the remaining positions are occupied by CCN-based routers. All links operate with bandwidth of 10 Mpbs and delay of (10 + r) ms where r follows a uniform distribution and can assume values in the interval [1,2] ms. On average, r = 1.5 ms.

**Rocketfuel:** The Rocketfuel topology, shown in Figure 2, is based on the topology of the ISP Exodus. This topology was obtained through the topology mapper Rocketfuel [6]. The network contains 192 nodes, where 95 are leafs, 58 are gateways and the remaining 39 comprises the backbone. During the simulations, the position of the publishers and the consumer were randomly chosen from the set of leaf nodes.

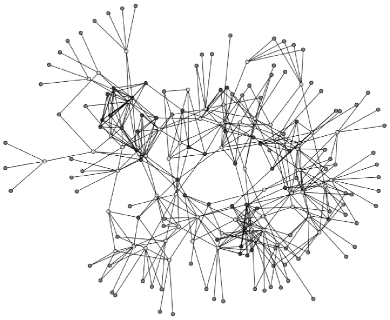

Fig. 2. Rocketfuel Topology

### B. Nodes' Behavior

**Publishers:** All legitimate publishers are associated to a name prefix. When an interest packet is received, the publisher compares the requested content's name with its own prefix. If a match is found, then the publisher sends a data packet back to the consumer. Otherwise, the publisher simply drops the received interest. On the other hand, malicious publishers are not associated to any prefix. All received interests are promptly answered with a polluted content. Both the legitimate and polluted contents are 1024 bytes. The difference is that polluted contents have a fake signature.

**Consumers:** In all the simulations executed, the consumer always requests 20 different contents at a rate of 10 interests/s. Each time the consumer requests a content, a timer associated with the content name is initialized. If the timer expires before a legitimate copy of the requested content is received, then the consumer will retry to request the content. For each content, the consumer is allowed to retry at most ten times before giving up to retrieve it. When a polluted content is received, the consumer verifies its signature and immediately discards it. In this case, no data structure is changed and the consumer keeps waiting for a legitimate copy. Consequently, in the consumer's point of view, the receipt of polluted contents is equivalent to receiving no content at all.

**Routers:** All routers implement `CCNCheck` and are configured with the same static signature verification probability. In addition, the routers are also configured to use the flooding strategy, in which an interest is forwarded through all available interfaces, except the interface from which the interest was received.

### C. Methodology

For the simulations on the Grid topology, for each value of p considered it was executed 500 simulation rounds. For the simulations on the Rocketfuel topology, 50 triples in the form (malicious publisher, legitimate publisher, consumer) were randomly built. Each field on the triple represents the position of a leaf node on the network in which the correspondent entity will be placed. For each triple, it was executed 500 simulation rounds. This process was repeated for all values of p considered.

### D. Metrics

CCNCheck aims to benefit not only the CCN users, but also the network. From the users perspective, CCNCheck will be effective if it increases the fraction of requested contents that return legitimate copies. On the other hand, the network will benefit from CCNCheck, if it reduces the amount of polluted contents being forwarded on the network. To evaluate these criteria, two metrics are proposed:

**Legitimate Fraction (LF):** fraction of requested contents that returned legitimate copies; and

**Polluted Messages (PM):** number of times that a polluted content is forwarded in the network.

## IV. RESULTS

In this section we evaluate the results achieved during the simulation experiments. All routers were configured with the same signature verification probability ($p$). Only values of $p$ up to 10% were considered in order to keep the verification overhead as low as possible.

Figure 3 shows, on average, how many requested contents were correctly (not polluted) retrieved when $p$ varies for both topologies. When $p = 0$, `CCNCheck` is disabled and nearly $50\%$ of all requested contents are never correctly retrieved. As we increase $p$, the curves representing the two topologies become very distinct. When $p = 10\%$, `CCNCheck` allows almost $90\%$ of the requested contents to be received correctly

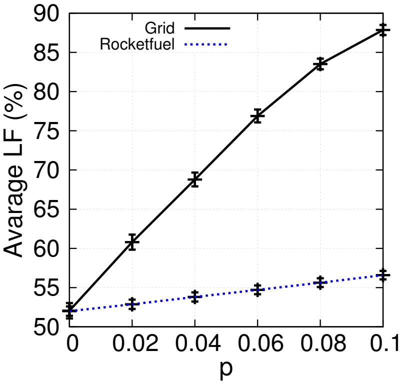

Fig. 3. Fraction of requested contents that was correctly (not polluted) received on the Grid and Rocketfuel topologies. All routers on the network verify signatures with the same value of $p$. The error bars were calculated with 95% of confidence.

on the grid topology, which represents an increase of about $40\%$. On the other hand, CCNCheck's efficiency on the Rocketfuel topology was much less expressive, allowing an increase of less than $10\%$.

This situation can be explained by the difference on the number of hops in the paths that connect the consumer to the malicious publisher on each topology. Because the grid topology considered on the simulations has dimensions 21x21, all paths that connect the consumer to the malicious publisher will have at least 20 hops. On the other hand, the Rocketfuel topology presents a lower number of hops on the paths between any two leaf nodes. Consequently, configuring all routers with the same value of $p$ makes the CCNCheck's efficiency dependent on the network topology.

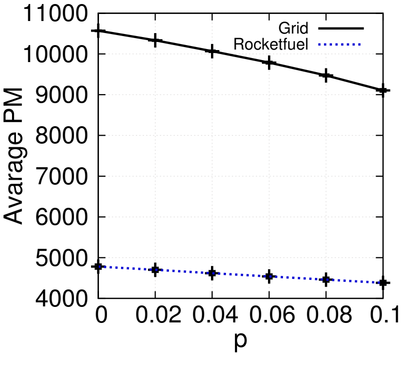

Fig. 4. Total amount of polluted messages forwarded on the Grid and Rocketfuel topologies. All routers on the network verify signatures with the same value of $p$. The error bars were calculated with 95% of confidence.

The same behavior can be observed in Figure 4, that shows the number of polluted messages forwarded on the network. When $p = 10\%$, CCNCheck provided a reduction on the PM metric of about $14\%$ on the grid topology and $8\%$ on the Rocketfuel topology. Although the CCNCheck's efficiency depends on the network topology when all routers have the same value of $p$, the mechanism allowed the users to retrieve more legitimate contents and reduced the number of polluted contents being forwarded on the network.

## V. Conclusions

In this work we presented CCNCheck, a mechanism that performs probabilistic signature verifications in order to mitigate the content pollution dissemination in Content-centric Networking. The benefits brought by the mechanism are twofold. First, it reduces the dissemination of polluted contents, increasing the availability of their legitimate counterparts. Consequently, users are allowed to retrieve more legitimate contents. Second, when the signature of a polluted content is verified, it is discarded and not forwarded. Avoiding forwarding polluted contents allows the network to spend its resources in processing and forwarding useful data, instead of wasting its resources processing and forwarding useless polluted data.

Simulation results show evidences that CCNCheck is able to increase the delivery of legitimate contents and to reduce the number of polluted contents forwarded in the network for both evaluated topologies. However, because all routers verify signatures with the same static probability, the mechanism's efficiency depends on the network topology. In future work we intend to evaluate solutions to reduce such dependency. For instance, we could enforce that border routers check signatures with higher probabilities than the routers in the core of the network if the traffic load at the border routers is lower then at the core of the network.

## Acknowledgment

This work is supported in part by CNPq, CAPES, FAPERJ, TBE/ANEEL, INCT-WebScience and CELESC/ANEEL.

## References

[1] "NS-3 Simulator," http://www.nsnam.org (Visited in 06/2014).

[2] I. M. A. Afanasyev and L. Zhang, "ndnSIM: NDN simulator for NS-3," NDN, Tech. Rep. NDN-0005.

[3] P. G. A. Compagno, M. Conti and G. Tsudik, "NDN Interest Flooding Attacks and Countermeasures," in *ACSAC, 2012*.

[4] I. M. G. Brito, P. Velloso, *Information Centric Networks: A New Paradigm for the Internet*. Wiley, 2013.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, and M. F. Plass, "Networking named content," in *CoNEXT, 2009*.

[6] R. M. N. Spring and D. Wetherall, "Measuring ISP topologies with rocketfuel," in *ACM SIGCOMM, 2002*.

[7] E. U. P. Gasti, G. Tsudik and L. Zhang, "DoS & DDoS in Named-Data Networking," in *ICCCN, 2013*.

[8] B. R. S. Arianfar, T. Koponen and S. Shenker, "On Preserving Privacy in Content-Oriented Networks," in *ACM SIGCOMM Workshop on Information-Centric Networking, 2011*.

[9] G. T. S. DiBenedetto, P. Gasti and E. Uzun, "ANDaNA: Anonymous Named Data Networking Application," in *NDSS, 2012*.

[10] T. Schwetzingen, "Security & Scalability of Content-Centric Networking," Master's thesis, Technische Universitat Darmstadt, 2010.

[11] D. Smetters and V. Jacobson, "Securing Network Content," PARC, Tech. Rep. TR-2009-1.

[12] A. K. Y. Gao, L. Deng and Y. Chen, "Internet cache pollution attacks and countermeasures," in *IEEE ICNP, 2006*.