Distributing Key Revocation Status in Named Data Networking

Giulia Mauri, Giacomo Verticale

Distributing Key Revocation Status in Named Data Networking

Giulia Mauri and Giacomo Verticale

Department of Electronics, Information, and Bioengineering, Politecnico di Milano
{gmauri,vertical}@elet.polimi.it

1 Introduction

Content Centric Networking (CCN) [1] is a new network paradigm designed to satisfy user needs considering the growth of data demand. Named Data Networking (NDN) [2] is a research project that is developing the future Internet architecture using the principles behind CCN. In this novel architecture, the contents are addressed by their name and not by their location. Thus, the attention is shifted from user to content, resulting in a caching network that is more efficient and flexible than an IP network for content distribution and management with beneficial effects on timely delivery. In NDN, the content objects are divided into chunks, each digitally signed by its producer, and most papers assume that verification is made only by the content consumer. In order to perform signature verification, a node needs the signer's key, which can be easily retrieved by issuing a standard interest message. Although content verification at the end node prevents disruptive attacks in which false data is delivered to applications, the verification of key validity is also necessary. Otherwise, false data would be cached and forwarded instead of correct data resulting in a denial of service and paving the way for more sophisticated attacks.

Indeed, content signed with a compromised key may remain in caches for an indeterminate amount of time, and possibly be served to end users. Even if caches implement a freshness mechanism that deletes content that has been in the cache longer than a given threshold, a compromised node could resend data making it extremely difficult to remove from the network the objects signed with a compromised key. In the standard PKIX (Public Key Infrastructure Certificate X.509) [3], the issue of delivering key revocation status to the end nodes is solved by using the OCSP (Online Certificate Status Protocol) protocol [4].

This paper proposes a way to implement similar functionality of OCSP protocol in the NDN scenario. In particular, we suggest three alternative ways to guarantee key freshness proposing two reactive methods and comparing them with a proactive method that is based on the ccnx-repository synchronization protocol. Finally, we use the open-source ndnSIM [5] package to run simulations for different network scenarios. We evaluate the performance in terms of latency, throughput and hit ratio gained by the proposed methods.

2 Distributing Key Revocation Status

Attacker Model

Our attacker model assumes an active eavesdropper, a *malicious non-intrusive* attacker, with the following properties:

- It behaves as a legitimate node of the network and communicates with any other node. In particular, it can send interests for contents and receiver the corresponding data packets;
- In response to an interest message it can deliver content packets created on the fly and signed with any signature for which it knows the corresponding signing key or with an invalid signature;
- In response to an interest message it can reply any content it knows even if the corresponding signing key is known to be compromised.

We observe that the attacker cannot (1) modify other nodes' routing tables and (2) break any cryptographic algorithm.

The attacker purpose is to inject bad contents into the end-node caches. Particularly, the saboteur could reach his goal in two ways:

1. responding to Interests with corrupted Data packets;
2. signing false packet with a revoked key.

We consider a packet *corrupted* if its signature is invalid, while *false* if its signature is valid but generated with a compromised key.

We state that the protocol is **secure** if it is not possible that a honest end-node (we call it consumer or leaf node) accepts a corrupted or false packet in its cache or delivers it to its neighbors.

Our proposal

Since the attacker has the ability to retrieve and use old keys to sign messages, in this section we provide three methods to guarantee that in a predefined time interval the key has not been revoked.

In the following paragraph, first we present two reactive modes based on a CCNx approach and then we report a proactive mode based on CCNx synchronization protocol.

Nonce-based In this method, we guarantee the up-to-date status of the key assuring that the key would be sent directly by the possessor. Particularly, the requesting node sends an Interest for the key specifying the `AnswerOriginKind`, that is 0, and it means "do-not-answer-from-content-store". The node expects to receive a Data packet containing the key and signed with the root key.

TimeStamp-based In this case, the node sends an Interest for the key to all its neighbors specifying in the name a timestamp that indicates a threshold validity. When a node receives the Interest, it checks if it has the key and if its key TimeStamp is more recent than the Interest TimeStamp, i.e. $TS_I < TS_D$. If the condition is satisfied, the node sends the corresponding Data packet containing the key. Otherwise, the node forwards the Interest to the following node. If no node has the fresh key, the Interest is forwarded till the original key possessor.

Proactive Mode As comparison, we consider the proposal of [6] and we adapt it for our purpose. All the key names have the conventional prefix `"/keys"` in order to facilitate the key management. The keys stored in ccnx-repositories are revoked, that is all the keys that are no longer valid to sign a packet. The repositories are synchronized using ccnx-sync protocol. Particularly, the root node defines the *Revoked Key Collection* where it can store the keys, then a SyncAgent is responsible for keeping Repository up to date as some changes happen. Every Δt seconds, where Δt is a time interval longer than the timestamp of the previous section, the synchronization between root and router repositories is performed using a RootAdvise message.

Security Evaluation We suppose indispensable in the content-centric scenario that each end-node should follow the signature verification process for each data packet and that it should check the key freshness. Moreover, we hypothesize that an attacker can only obtain or find invalid keys. In other words, we guarantee that a node doesn't accept a corrupted or false packet in its cache or delivers it to its neighbors within a time interval longer than Δt, that depends on freshness method used. Thus, we can say that the network is **secure** under these assumptions.

2.1 Numerical Results

In this subsection, we present simulative results to evaluate which is the impact on network performance for implementing key retrieval. Especially, we report only data concerning delay, due to space constraints, considering the different choices that a node can make for the freshness method.

In Figure 1 are depicted the mean round trip times to obtain data and key as a function of content class (the lower is the class, the higher is the popularity). Results are reported for the first 20 classes of popularity. The round trip time is the interval of time that incurs from the first interest sent for a data till the corresponding key reception. The delay is highly correlated to the distance between user and content and also to the content popularity. Moreover, the latency depends on the freshness method used: the Nonce-based method requires more time than the others because the key is asked more frequently and it comes from the root, that is the farthest node; the proactive mode is the faster method, since the revoked key repositories are synchronized off-line; while the TimeStamp-based approach stays in the middle and it depends also on time

interval length of key validity. We note that increasing the key validity interval, the delays of proactive mode and timestamp methods coincide.

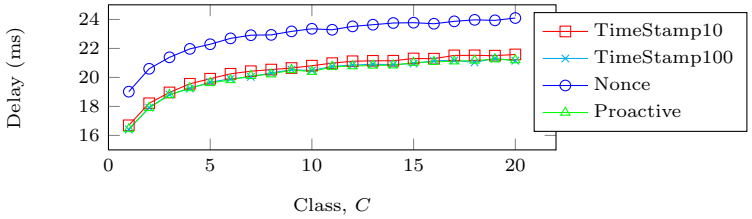

Fig. 1. Mean latency related to popularity classes in tree topology considering alternative freshness methods

Observing the results, we think that the timestamp-method not only prevents nodes accepting a corrupted or false packet in their cache but also achieves the better compromise in terms of delay, throughput and hit ratio. Moreover, we believe that the proactive mode can be used when the channel capacity is limited, since the revoked key update is done off-line. While, the nonce-based method guarantees that the keys have a shorter interval of uncertainty but we have to introduce more traffic on the network and also wait more time for each content.

3 Conclusion

Whenever a cryptographic key is revokated, the nodes should reject all the contents signed with that key and these packets should be removed from the network. This paper compares three possible methods to achieve this result. In the reactive methods, the consumer nodes request status certificates for the keys they need. The freshness of these certificates can be verified by using nonces or timestamps. In the proactive method revocation lists are broadcast using the ccnx synchronization protocol. We compare the performance of content retrieval in terms of delay, throughput and hit ratio. We conclude that the solution based on timestamps provides the best compromise between delay and key status distribution overhead. This research represents also a first step into the problem of authenticity and integrity of contents.

References

1. V. Jacobson *et al.*, "Networking named content," in *Proceedings of the 5th CoNEXT '09*. New York, NY, USA: ACM, 2009, pp. 1–12.
2. L. Zhang *et al.*, "Named data networking (ndn) project," University of California and Arizona, Palo Alto Research Center and others, Tech. Rep., October 2010.
3. D. Cooper *et al.*, "Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile," RFC 5280, May 2008.
4. M. Myers *et al.*, "X.509 internet public key infrastructure,online certificate status protocol - ocsp," RFC 2560, June 1999.
5. A. Afanasyev *et al.*, "ndnsim: Ndn simulator for ns-3," UCLA, Tech. Rep., 2012.
6. C. Bian *et al.*, "Deploying key management on ndn testbed," UCLA, Peking University and PARC, Tech. Rep., Feb, 2013.