Content-Location Based Key Management scheme for Content Centric Networks


Published in:
Proceedings of the 6th International Conference on Security of Information and Networks

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
© Copyright 2013 ACM This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in Proceedings of the 6th International Conference on Security of Information and Networks, http://dl.acm.org/citation.cfm?doid=2523514.2523566

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.
Abstract—Content Centric Networking (CCN) is a promising routing paradigm for content dissemination over the Internet of the future based on named data instead of named hosts. The CCN architecture will enable more scalable, secure, collaborative and pervasive networking. In particular, the CCN security scheme relies on the authentication of every transmitted content; which must be signed. However, this introduces new challenges regarding the authentication of content through the management of encryption keys linked to content providers. To keep the privacy and integrity of content using encryption, a proper key management scheme is necessary to authenticate the encryption key as well. In this paper, we propose an online public key generation technique which is suited for the CCN architecture. We analyze the proposed technique using the ccnSim simulator to evaluate its performance and the AVISPA tool to assess its security. Besides specific features, our scheme shows better performance in terms of time taken by a node to retrieve and generate a public key for the received content compared to the standard PKI approach.

I. INTRODUCTION

CCN was Developed at PARC by Van Jacobson and his team [1]. Content Centric Networking (CCN) is also known as Information Centric Networking or Named Data Networking [2]. Now-a-days ost of the communications are based on content retrieval (web, P2P, etc.) than point-to-point communications (VoIP, IM, etc.), hence CCN communication architecture is the best match to its current usage. Content is addressable, routable, self-sufficient and authenticated, while locations no longer matter. Data is seen and identified directly by a routable name instead of a location (the address of the server) and is directly requested at the network level, not its provider. To improve content diffusion, CCN relies on close data storage because storage is proven cheaper than bandwidth: every content - particularly popular one - can be replicated and stored on any CCN node, even untrustworthy. People looking for particular content can securely retrieve it in a P2P-way from the best locations available.

CCN has two main types of packets, Interest and Data. A user who wants to access a given content sends out an Interest packet, specifying the name of the content (as defined by CCN nomenclature ContentName) to all its available faces. Faces can be anything which can serve as medium for transmitting and receiving data. A node which receives this packet and that can ”satisfy” the Interest then sends out the corresponding Data packet onto the face from which it received the Interest. By definition, CCN nodes are stateful and only forward Data on the back path if an Interest was emitted beforehand. Consequently, CCN data cannot be sent toward a node without any prior Interest request from that node (Data cannot be pushed, only pulled).

In this paper, we propose a distributed key management scheme that could fit the needs of CCN. After presenting the related work on the security of CCN and on some existing key management schemes in Section III, we describe our key management scheme in Section IV before evaluating it under both a performance point of view in Section V and a security point of view in Section VI. Finally, Section VI concludes and presents our future work.

II. RELATED WORK

If CCN improves security in some points, it also raises the possibility of new kinds of attacks. For example, CCN routers are more vulnerable than IP routers because of their memorable nature and the complexity of the management of their inner tables, which are essential for performance and quality of service. All these new algorithms and their implementations need to be assessed from a security point of view before being trustable.

Due to memorable routers network devices are exposed to new threats that can be exploited for attacks. Attackers may disturb the management of CCN node’s tables which are necessary to ensure performance and quality of service. This can lead to new possible DoS (Denial of Service) attacks that must be detected and mitigated.

Tobias Lauinger [3] identified several attacks related to caches, in particular denial-of-service attacks against CCN routers, but he only investigated one of these, ”cache snooping” that enables attackers to efficiently monitor the content retrieved by their direct neighbors. Finally, Wählisch et al. [7] identified possible attack scenario ranging from classical resources exhaustion to more complex injections of malicious names or routes due to the capacity of end users to directly alter the routing plane.

To improve CCN security, Smetters et al. [4] of PARC propose authenticating the links between names and content, in addition to names and content themselves, in order to identify trustworthy content and avoid malware diffusion. However, the necessary Public Key Infrastructure (PKI) is hard to set up and if a provider becomes malicious, revocation remains a major problem that must be addressed in future work.
Since little investigation has been made in securing CCN, there is no specific key management scheme for that. Here we describe some of the existing key management schemes proposed in the literature for the static and mobile ad hoc networks which is a field of networking where key dissemination in the network has been investigated for years.

In symmetric key cryptography, every node in the network is assigned $N-1$ keys where $N$ is the total number of nodes in the networks. This is not suitable for a large networks because each node is required to store large number of keys. In asymmetric cryptography, each node is assigned a pair of keys (i.e., public key and private key) for secure communication with other nodes in the network. Recent research works in cryptography are mainly based on the traditional public key infrastructure (PKI [5], [6], [7], [8]), and identity based public key cryptography (ID-PKC [9], [10]). Here we discuss the compatibility of those approaches with CCNs.

Smetsers in [4] suggested the standard PKI approach for CCNs. In this approach, each node has a pair of keys (public key and private key) and for secure communication, each node publishes its public key along with its certificate, assigned by a certification authority. Each node in the network verifies the public key of other nodes by sending the attached certificate to the certification authority for validation. The certification authority check the validity of the certificate and sends the acknowledgement back to the node about the authenticity of the certificate. Since PKI is based on the concept of a single centralized certification authority, it does not suit well the concept of CCNs, where contents are replicated in the network and are requested and routed by name instead of being provided by a single source. This increases the number of public/private key pairs, which in turn increases the verification overhead for the single certification authority.

In 2004, Deng proposed a new cryptography management and certification scheme called ID-PKC [10]. In this scheme, there is a master public/private key pair. The master public key is known by all nodes in the network, while the master private key is divided into shares and distributed among $k$ nodes of the network (fewer than the total number of nodes). Each node ID is working as node public key and for secure communication, it needs its private key. So the node sends a request to $k$ PKGs (Public Key Generators) to get its private key share. The requesting node generates a temporary public/private key pair and gives that public key to the PKGs to get its encrypted private key share. After getting those shares, the node generates its private key. Then the PKGs announce the requesting node ID to all nodes of the network, to be used as its public key. This scheme, however, still does not address the problem of updating the main key of the system.

III. PROPOSED KEY MANAGEMENT SCHEME

Key management schemes for TCP/IP networks secure source to destination link irrespective of the type of packets/data. Hence these schemes are ill suited for the CCNs architecture, where there is no concept of link between source and destination.

Standard PKI approach is based on centralized certification authority architecture and when the destination node receives the public key of the source node, it validates the received key from the certification authority. Also a node uses its private key to encrypt all the content and the destination node uses the public key with the certification authority to verify the received contents. But content centric networks architecture is based in the content itself, hence each received content and its key should be verified from the certification authority, which increases the overhead. Hence in this paper we propose to use the idea of using key shares to check the authenticity and integrity of the decryption key as well as of the received content.

We propose an authentication and key establishment scheme for CCNs in which the keys are related to the contents and their current location. However these keys are generated using key generation shares based on the contents and their location. The content requesting node can get those shares from any node in the network, even from untrustworthy ones, in accordance with a key concept of CCNs. In our work we also provide means to protect the distributed shares from modification by these malicious/intruder nodes.

A. Network Architecture

The internet is a composition of a large number of small networks i.e. WANs which have centralized controlling server which for management and security-related aspects of networking.

Since each small network consists in a large number of nodes, it is assumed that some of the are selected by the main server as its immediate one hop neighbors. It is done to make sure that all the traffic going into the network first passes through these selected nodes. These selected nodes are responsible for key share generation based on their network location. These nodes are called Location Based Key Share Generators (LBKSG). Remaining nodes are normal nodes (NN). Both types of nodes are similar in terms of capabilities and architecture.

Each node (both LBKSG and NN) is also assigned some key materials to generate their public/private key pair for securing the generated content. The assignment of those key materials to the nodes is performed off-line. The network manager also plays an important role in generating Location Based Key Share (LBKS) by providing unique identification code (UIC) to each LBKSG based on their physical location.

B. Key Material Assignment

Each node in the network is assigned some important key materials which are used in generating the public/private key pair in order to secure the generated contents. More specifically,

- Each node in the network is assigned a random number generator, a one way Hash function (H), a Location Based Key Share Generation function (f), a group generator G
- Each network server has also a fixed random number (FRN), a one way Hash function (H), a Location Based Key Share generation function (f), a group generator G.
C. Key Establishment and Management

After the network deployment, each node sends a join message to its network server. After the reception of those join messages, the network server sends a UIC and a Network Server Share (NSS) to each joining node. It is generated as

\[ NSS = f(FRN, \text{Server UIC}, \text{Node ID}, \text{Node UIC}) \]  

When a node receives the Node-UIC and NSS from server, it generates a Node Location Share (NLS) of its public key as

\[ NLS = f(\text{Node UIC}, \text{NSS}) \]

The required public key is

\[ k_{\text{pitc}} = (NLS + NSS) \mod G \]

Where its related private key is

\[ k_{\text{prt}} = k_{\text{pitc}}^{-1} \mod G \]

In CCNs, since contents are requested by their names instead of their generating source, there must be a relationship between the content and its encryption/validation key. Hence we introduce this relationship by relating the content with the content holding node location.

Whenever a node request for a content, it will also get NLS from the content holding node. If the content holding node is actual content generating source, even then it will send its NLS to the requesting node.

Once the requesting node get NLS and content from a node, it requests its server to get the NSS of the content. Its server requests the content holding node’s server for the NSS. After receiving the NSS, it sends it back to the requesting node. After the successful verification, the requesting node updates its NLS to the requesting node.

Once the requesting node get NLS and content from a node, it requests its server to get the NSS of the content. Its server requests the content holding node’s server for the NSS. After receiving the NSS, it sends it back to the requesting node. After the successful verification, the requesting node updates its NLS to the requesting node.

IV. Performance Analysis

In order to evaluate the performance of the proposed scheme against the standard PKI approach, we use different network topologies provided in the ccnSim simulator [11], developed specifically for content centric networks, and note the average time taken by a node to retrieve the key(s) for the received content(s). To do so, each node in the network generates a content which is composed of 100 files. Each file is encrypted by a separate key and the corresponding key shares are distributed among the LBKSG. Also each file is split into five chunks. When a node start receiving the requested file after sending an interest for that file, it waits until all the chunks of the requested file arrive. Once the requested file is completely received, the node (requester) sends a request for the key shares of the received file. After the reception of those key shares from the nearest LBKSG (all others will be discarded, according to the CCN principle), the requester verifies the authenticity and integrity of the received file.

A. Results

During the simulation, we selected the network server for holding the key related shares based on location for the standard PKI approach while we selected three first single hop neighbors to hold and generated the key share for the received content based on their location. Figure 1 shows the average time taken by a node in different network topologies to retrieve a key for the received content. Our scheme is at least as good as a central authority and up to 50% faster to retrieve the key in some specific topologies.

![Fig. 1. Virtual organization of Key Holding Nodes and Normal Node in the network](image_url)

V. Security Analysis

Since we kept a relationship between the content and its encryption key using the current location of the LBKSG, only the authentic LBKSG of the network are able to generate the key shares and encrypt the content(s) using that key(s). An adversary would not be able to encrypt the content using the authentic generated key until and unless it becomes an authentic member of the network (i.e. it receives the key material described above before deployment). On the other hand, by using the standard PKI approach for CCNs, an adversary can generate and encrypt fake content using its generated private key corresponding to an authentic public key. This is easy because in CCNs key request is based on the content name instead of the generating source.

Centralize certification authority requirement for the verification of the generated keys has been eliminated in the proposed scheme by using the key share concept based on content and its location.

In order to validate the secrecy of the proposed key management scheme, we used the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool [12]. AVISPA is a push-button tool for the automated validation of Internet security-sensitive protocols and applications. It provides a modular and expressive formal language for specifying protocols and their security properties, and integrates different back-ends that implement a variety of state-of-the-art automatic analysis techniques (e.g. OFMC, ATSE, etc).

We implemented the proposed key management scheme in AVISPA and checked its security using some of the attacks provided by AVISPA, namely OFMC (On-the-Fly
Model-Checker) and CL-AtSe (Constraint-Logic-based Attack Searcher).

The former builds the infinite tree defined by the protocol analysis problem in a demand-driven way, i.e. on-the-fly and uses a number of symbolic techniques to represent the state-space. The latter provides a translation from any security protocol specification written as transition relation into a set of constraints which can be effectively used to find attacks on protocols. Both translation and checking are fully automatic and internally performed by CL-AtSe, i.e. no external tool is used. In this approach, each protocol step is modeled by constraints on the adversary knowledge. These results are shown in table I and tends to prove that our solution does not suffer from security issues.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFMC</td>
<td>SAFE</td>
</tr>
<tr>
<td>CL-AtSe</td>
<td>SAFE</td>
</tr>
</tbody>
</table>

**TABLE I**
AVISPA SIMULATION RESULTS

Content centric networks basic concept has been followed in the proposed scheme. According to this, each node can get the required key shares and contents from any node of the network, even an intruder/attacker. This means that the key management scheme must be so secure that the intermediate nodes/attackers cannot modify the contents/keys which are provided by our proposed scheme. Even if a node gets the required shares from the intruder who acts as a man-in-the-middle, a node is able to verify and authenticate those shares.

**VI. CONCLUSION**

In this paper a key management scheme for Content Centric Networking based on the content and its current location has been proposed. As this new networking paradigm heavily rely on content authentication to create a new web of trust, an efficient key management scheme is required for the further development of CCN. In proposing our key management scheme, we took care of these two constraints by: (1) introducing LBKSG to reduce the communication overhead used for authentication and (2) key shares to check the authenticity and integrity of the received content. Performance analysis on ccnSim showed that our scheme is at least as good as a central authority while its distributed nature make it more scalable. Thus, the security analysis performed thanks to avispa tends to prove that no security leak exists.

**REFERENCES**


