A Key Management Scheme for Content Centric Networking

Sarmad Ullah Khan, Thibault Cholez, Thomas Engel, Luciano Lavagno

A Key Management Scheme for Content Centric Networking

Sarmad Ullah Khan, Thibault Cholez*, Thomas Engel*, Luciano Lavagno
Department of Electronics, Politecnico di Torino, Italy
*SnT, University of Luxembourg, Luxembourg
Email: {firstname.lastname}@polito.it; *{firstname.lastname}@uni.lu

***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

Developed at PARC by Van Jacobson and his team [1], CCN is also known as Information Centric Networking or Named Data Networking [2]. Building on the observation that today's communications are more oriented towards content retrieval (web, P2P, etc.) than point-to-point communications (VoIP, IM, etc) CCN proposes a radical revision of the Internet architecture switching from named hosts (TCP/IP protocols) to named data. Content is addressable, routable at the network level, self-sufficient and authenticated, while locations no longer matter. A user who wants to access a given content sends out *Interest* packets, specifying the name of the content. The *Interest* is then forwarded up to the closest node which can answer with a *Data* packet.

CCN has a built-in security layer to ensure that the content received by a previously announced Interest is authentic. As in CCN only the content matters but not the route it takes, the only thing which needs to be checked for authenticity, consistency and integrity is the content itself. The key-stone of CCN security is the trust in the publisher. Content can be authenticated by every node using public key signatures. To securely authenticate content, CCN has to bind the content name, the content itself and the content provider. Every piece of data must include a way to retrieve the key of publisher and mapping evidence. To do so, the following information is embedded in each CCN data packet: $Signature(Name, Content, SignInfo)$ with *SignInfo* including: a cryptographic digest or a fingerprint of publisher's key, the key or the key location. To improve CCN security, in [3], Smetters et al. propose to authenticate the links between names and content with the following rule: each new content creates a mapping triple: $M_{N,P,C} = (N, C, Sign_P(N, C))$ where M is the Mapping, N the Name, P the provider, C the Content.

When a node receives the key of a provider, it must have a way to verify it (for example, using a certification authority in the Public Key Infrastructure approach). However, the necessary PKI is hard to set up and would lead to scalability issues. The nodes should be able to generate the key material to certify their Data and retrieve the key material from others to authenticate the received Data. To be scalable, the system should also provide a way to authenticate the encryption keys as well. Therefore, in this paper, we propose a distributed key management scheme that could fit the needs of CCN.

II. Related work

So far, there has been little investigation of the security of CCN and we currently lack knowledge about possible security issues. If CCN improves security in some points, it also raises the possibility of new kinds of attacks. In particular, due to stateful routers (as the route between a content and the requester has to be memorized) network devices are exposed to new threats that can be exploited for attacks. Tobias Lauinger [4] identifies several attacks related to caches, in particular denial-of-service attacks against CCN routers, but he only investigates one of these, "cache snooping" that enables attackers to efficiently monitor the content retrieved by their direct neighbours. Concerning the key management, Jacobson et al. [1] propose to use the SDSI/SPKI where keys are mapped to identities via namespaces (CCN names) so that there is no single source of trust like the current certification authorities. Key management is still an open issue., several solutions are cited which range from a PKI to PGP like web-of-trust.

Concerning key management schemes, 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-PKG)[9],[10], ID-PKG completely eliminates the need for public key certificates by exploiting publicly known user identity information (such as IP address or telephone number) as a public key for securing

information. ID-PKG enables any pair of users to communicate securely without exchanging public key certificates, and without using the online services of a third party. In 2003, the first ID-PKG cryptography management and certification scheme [9] for mobile ad hoc networks was presented by Khalili and Katz. The basic idea of the scheme is similar to the scheme of Zhou and Haas [5].

In 2004, Deng proposed a new cryptography management and certification scheme called ID-PKC [10] which has 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 used as node public key, and each node generates its private key using private key shares obtained from k Public Key Generators (PKGs) by using a temporary public/private key pair. This scheme, however, still does not address the problem of updating the main key of the system.

III. Architecture of the Key Management Scheme

The existing key management schemes for TCP/IP networks secure the links from source nodes to the destination nodes irrespective of the number and type of packets/data. Hence these schemes are ill suited for the CCNs architecture, where there is no concept of link between the requesting node and content generating node. Although the standard PKI approach, where the receiver verifies the received public key of the sender through the certification authority (CA), is being in consideration for CCN, this approach is not well suited to the CCN concept where the keys are related directly to the content, instead of the sender ID or its location. Indeed, in the standard PKI approach, a node uses its encryption key (private key) to encrypt all the content, and the destination node needs to verify the decryption key (public key) with the certification authority to check the authenticity and integrity of all the received contents, which means that a key belongs to a specific node. But in Content-Centric networks, there is no concept of content source ID/location information. Hence in this paper we propose to use: (1) distributed key holding nodes to reduce the communication overhead on a single node and (2) key shares to check the authenticity and integrity of the decryption key as well as of the received content.

In the proposed scheme, we keep a relationship between the content and its encryption key using the NeTwork Public Share (NTPS) and NoDe Public Share (NDPS) (recognized by the content) so that only the authentic nodes of the network would be able to authenticate the content(s) using the authentication key(s). The content requesting node can get those shares from any node in the network, even from malicious and intruder ones, in accordance with a key concept of CCNs.

Also we eliminate the need for a central certification authority for the verification of the encryption keys by using the key share concept and by securing the keys with the network and node parameters. So the adversary would not be able to generate a fake private key simply thanks to the non-existence of a single complete public key.

The proposed scheme hence follows the basic concept of Content-Centric networks, which states that a node can get the required authentication keys 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.

A. Network Architecture

Since the Internet is a composition of large number of small networks, we assume that each individual network has its own network manager. These are powerful secure nodes which act like servers for security-related and other aspects of networking.

Each small network consists of a large number of nodes. We divide all the nodes of each small network into two different categories, i.e. (1) Normal Nodes (NN) and (2) Key Holding Nodes (KHN). Both types of nodes are similar in terms of their capabilities and architecture. The KHNs are responsible for initially holding the key materials of the encryption key(s) related to the content(s) once they are generated by any node (source) of the network after the network deployment.

The selection of KHNs is based on the maximum number of connections established by a node with its neighboring nodes. This approach minimizes the initial network traffic due to key management, since each node will be at a maximum of one hops from a KHN. Note that when nodes are deployed they are all assigned the same security-related material and hence can play both roles. The distinction between KHN and NN is made after deployment, when the node joins a network, and only affects the role that the node plays in providing and using the key material.

In order to select KHNs, each node shares its connection count with its neighboring nodes. Once all the neighboring nodes receive those counts, each node selects a neighboring node with the highest number of connections as its nearer KHN. Also if two neighboring nodes have the same connection count, they both become KHNs for each other and to their less connection count neighboring nodes.

Figure 1 shows the virtual organization of KHN and NN in the network. It should be clear from the figure that nodes 3, 4, 8 have the highest number of connections with their neighboring nodes in the network, so they act as actual KHNs. On the other hand, nodes 2, 5, 7 and 9, on one side, connected to KHNs directly and on the other side connected to the node have less number of connections and are not connected to the actual KHNs directly, so they act as virtual KHNs for them. For example, node 2 will act as a virtual KHN for node 1 while node 1 will act as virtual KHN for node 15. This approach defines a routing path to the actual KHNs in the network.

Each node (both KHN and NN, since also the former can generate contents) is also assigned some key material to generate their public/private key pair for securing the generated content. The assignment of those key material to the nodes is performed off-line while the assignment of content specific key material to the KHNs is performed on-line. The network manager also plays an important role in generating the key material for its network nodes.

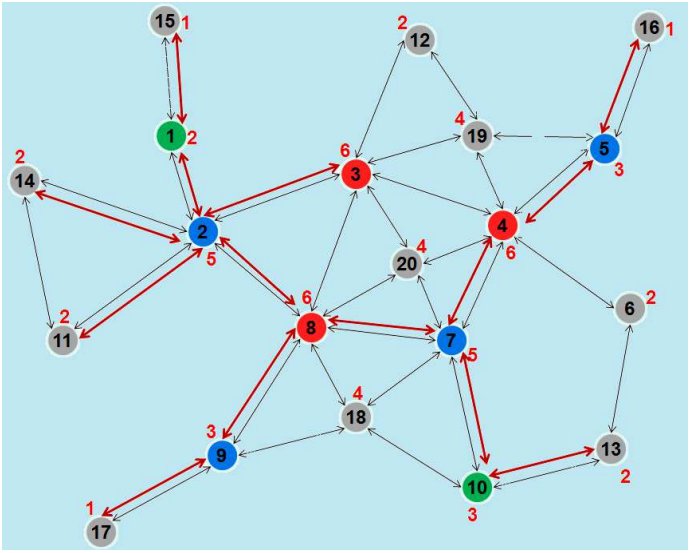

Fig. 1. Virtual organization of Key Holding Nodes and Normal Node in the network

B. Key Material Assignment

Each node in the network is assigned some important key material which is used for the public/private key pair generation 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 share generation function (f) and a natural number group generator G (G can be, for example, a prime number).
- Each network manager is also assigned a fixed random number (NMRN) assigned by the network owner, a random number generator, a one way Hash function (H), a share generation function (f) and a group generator G.

C. Key Establishment and Management

Once the network is deployed, each node in the network sends a join message to its network manager. After the reception of those join messages, the network manager sends a fix random number (NDRN) and a NeTwork Public Share (NTPS) to each joining node. The network public share is generated as:

$$NTPS = f(NMRN, node\ ID, RN) \tag{1}$$

Where RN is a random number from the generator. When a node receives the random number and NTPS from its network manager, it generates a NoDe Public Share (NDPS) of its public key as:

$$NDPS = f(NDRN, RN, NTPS) \tag{2}$$

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 two further shares generated by the source node of the content in order to relate the encryption/validation key with the content. Those two shares are P_1 and P_2 which act as the two parts of the public key for a content. These two shares (P_1,P_2) are generated as:

$$P_1 = f(NTPS + Content) \tag{3}$$

$$P_2 = f(NDPS + P_1) \tag{4}$$

The required public key k_{plc} is

$$k_{plc} = P_1 + P_2 \tag{5}$$

Since each node is given a group with a generator G, it selects a random number g from G and also creates the hash of the content-related public key shares and the corresponding private key k_{prt} as:

$$X = H(P_1), \quad Y = H(P_2), \quad Z = H(k_{plc}) \tag{6}$$

$$k_{prt} = K_{plc}^{-1}\ mod\ G \tag{7}$$

The node also calculates A, B and C for the authentication of the generated content and its shares as:

$$A = g^X, \quad B = g^Y, \quad C = g^Z \tag{8}$$

After the generation of the public key shares and their hashes, the node distributes those shares among the KHNs responsible for holding them $(i.e. P_1, P_2, C, Z, NDPS)$. The KHNs get the NTPS of the received NSPS from the network manager. The node includes A and B in the data packet along with the hash of content in order to help the destination node get and verify the public key shares i.e. $(Content, A, B, Z)$.

If now a node receives a data packet containing the content and hashes for the authentication, it needs the public key shares to verify those received hashes. In order to get the public key share from the KHNs, it sends a key share request to the KHN nodes. The KHN sends $(NDPS, NTPS, C)$ to the requesting node. After receiving this message, the node generates P_1 and P_2 using (3) and (4) and the share generation function f. After the generation of P_1 and P_2, the node generates X and Y using (6). Now the node calculates (C^X, C^Y) using the received C and calculated X and Y and then compares them with the (A^Z, B^Z) received in the data packet. Successful verification authenticates the received messages and the contained public key shares.

IV. Performance Analysis

In this section, we describe the performance of our proposed scheme for content centric networks in terms of time taken by a node to retrieve a key. We compare it with the standard PKI approach for key establishment and management. To this aim, we use the ccnSim simulator [11] developed specifically for Content-Centric Networking.

A. Simulation scenarios

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 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 Key Holding Nodes (KHNs). Also each file is split into five chunks. When a node starts 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 KHN (all others will be discarded, according to the CCN principle), the requester verifies the authenticity and integrity of the received file. Table I shows the average time taken by a node in different network topologies to retrieve a key for the received content.

During the simulation, we select one node as a key holding node for the standard PKI approach and three nodes as key holding nodes for the proposed scheme.

Scheme	Geant topology (s)	Level3 topology (s)	Tiger topology (s)	dtelecom topology (s)
PKI	0.009	0.020	0.0003	0.0133
Our	0.004	0.018	0.0002	0.0131

TABLE I
AVERAGE TIME TAKEN BY A NODES TO RETRIEVE A KEY FOR A CONTENT IN DIFFERENT NETWORK TOPOLOGIES

V. SECURITY ANALYSIS

Since cryptography is considered as the main building block of any security primitive, the keys must also be secured and authentic. To this aim, a key management scheme must be secure and each node of the network must be able to authenticate the key(s). This is the most challenging problem in CCNs, where the keys are linked with the content names instead of the content generation source. Hence we tried to solve the problem in our proposed key management scheme for the CCN networks. This would not be possible with the existing key management schemes for traditional TCP/IP networks.

In order to validate the secrecy of the proposed key management scheme for Content-Centric Networks, we used AVISPA (Automated Validation of Internet Security Protocols and Applications) [12]. AVISPA is a 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 II.

Technique	Summary
OFMC	SAFE
CL-AtSe	SAFE

TABLE II
AVISPA SIMULATION RESULTS

VI. CONLUSION

In this paper, a key management scheme is proposed for Content-Centric Networking. The scheme eliminates the need for a centralized key management authority without any compromise on the security level. Also the proposed scheme has the capability of authentication and verification of both the received data and the key information required to generate securely the required secret key. The proposed scheme showed better performance compared to standard PKI approach thanks to the use of distributed caches. Its security validation and testing has been performed using the AVISPA tool.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[2] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, kc claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named Data Networking (NDN) Project," October 2010.

[3] D. Smetters and V. Jacobson, "Securing Network Content," PARC, Tech. Rep., October 2009.

[4] T. Lauinger, "Security & Scalability of Content-Centric Networking," Master's thesis, TU Darmstadt, September 2010. [Online]. Available: http://tubiblio.ulb.tu-darmstadt.de/46912/

[5] L. Zhou and Z. Haas, "Securing ad hoc networks," *Network, IEEE*, vol. 13, no. 6, pp. 24 –30, nov/dec 1999.

[6] J. Kong, Z. Petros, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *Network Protocols, 2001. Ninth International Conference on*, nov. 2001, pp. 251 –260.

[7] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing ad hoc wireless networks," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, july 2002, pp. 567 – 574.

[8] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *Mobile Computing, IEEE Transactions on*, vol. 2, no. 1, pp. 52 – 64, jan.-march 2003.

[9] A. Khalili, J. Katz, and W. Arbaugh, "Toward secure key distribution in truly ad-hoc networks," in *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*, jan. 2003, pp. 342 – 346.

[10] H. Deng, A. Mukherjee, and D. Agrawal, "Threshold and identity-based key management and authentication for wireless ad hoc networks," in *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, vol. 1, april 2004, pp. 107 – 111 Vol.1.

[11] D. R. G. Rossini, "Caching performance of content centric networks under multi-path routing (and more)," in *Technical report, Telecom ParisTech*, 2011.

[12] "http://www.avispa-project.org/."