Experimental Study of Security Architectures for Wireless Mesh Networks

Fabio Martignon
Department of Information Technology and Mathematical Methods
University of Bergamo, Italy
Email: fabio.martignon@unibg.it

Stefano Paris
Department of Electronics and Information
Politecnico di Milano, Italy
Email: paris@elet.polimi.it

***Abstract*—In this paper we demonstrate the potential of MobiSEC, a complete security architecture for Wireless Mesh Networks (WMNs), that provides both access control for mesh users and routers as well as a key distribution scheme.**

MobiSEC has been implemented and integrated in MobiMESH, a WMN implementation that provides a complete framework for testing and analyzing the behavior of a mesh network in real-life environments.

Our experimental testbed compares MobiSEC both with a static fixed-key approach and with an end-to-end solution that consists in establishing an encrypted IPSec tunnel.

Experimental studies show that MobiSEC considerably increases the WMN security, with a negligible impact on the network performance, thus representing an effective solution for wireless mesh networking.

***Index Terms*: Wireless Mesh Networks, Experimental Test-bed, Authentication, Security.**

I. Introduction

Wireless mesh networks (WMNs) have emerged recently as a technology for next-generation wireless networking [1].

The network nodes in WMNs, named mesh routers, provide access to mobile users, like access points in wireless local area networks, and they relay information hop by hop, like routers, using the wireless medium.

Security in WMNs is still in its infancy, as very little attention has been devoted so far to this topic by the research community [1], [2]. Although many security schemes have been proposed for wireless LANs and ad hoc networks [3], [4], they are not suitable for WMNs, which need convincing security solutions that should act as incentives for customers to subscribe to reliable services.

In this paper we show MobiSEC[1], a complete security framework originally proposed in [5] for both the access and backbone areas of the WMN. It provides access control for end-users and mesh routers as well as security and integrity of all data communications that occur in the WMN. This is achieved with layer-2 encryption that uses a shared key whose delivery is assured by two key distribution protocols.

MobiSEC extends the IEEE 802.11i standard to the WMN scenario, exploiting the routing capabilities of wireless mesh routers. A two-step approach is adopted: in the first step new nodes perform the authentication process with the nearest mesh router, according to the 802.11i protocol, like generic wireless clients. In the second step, these nodes can upgrade their role in the network, becoming mesh routers, by further authenticating to a central server, obtaining a temporary key with which all traffic is encrypted.

The proposed solution has been implemented and integrated in MobiMESH [6], a WMN experimental platform that provides a complete framework for analyzing, studying and testing the behavior of a mesh network in a real-life environment.

We measured the performance of MobiSEC in several realistic network scenarios and we compared it both with a static approach that consists in using a fixed key to protect the WMN, as well as with an end-to-end solution that consists in establishing an encrypted IPSec tunnel.

Experimental results show that MobiSEC exhibits very high robustness against cryptanalytic attacks, and it further obtains a very high throughput, which is almost the same as that achieved by the static key approach. Therefore, MobiSEC represents a very effective solution to provide security in WMNs without impairing the network performance.

The paper is structured as follows: Sections II describes the MobiSEC security architecture and the key distribution protocol. Section III discusses experimental results which show the effectiveness of our solution as well as the demo setting.

II. MobiSEC Architecture

In this Section we describe MobiSEC, the architecture we proposed in [5] to authenticate the mesh routers and secure the traffic exchanged over the wireless backbone of a WMN.

In MobiSEC, client security is guaranteed using the standard 802.11i protocol, while backbone security is provided as follows: each new router that needs to connect to the mesh network first authenticates to the nearest mesh router exactly like a client node, gaining access to the mesh network. Then, it performs a second authentication connecting to a Key Server able to provide the credentials to join the mesh backbone. Finally, the Key Server distributes the information needed to create the temporary key that all mesh routers use to encrypt the traffic transmitted over the wireless backbone.

Figure 1 shows the three phases of the connection process performed by a new mesh router (namely, node N_2). When N_2 wants to connect to the mesh network, it scans all radio channels to detect a mesh router already connected to the wireless backbone, which is therefore able to provide access to all network services (including authentication and key distribution). Let N_1 be such router. After connecting to N_1, N_2 can perform the tasks described by the IEEE 802.11i

[1]A Network Simulator (ns2) implementation of MobiSEC is available online at http://home.dei.polimi.it/paris/mobisec.html

protocol to complete a mutual authentication with the network and establish a security association with the entity to which it is physically connected (phase 1). In phase 2, N_2 establishes a secure connection with the Key Server (KS), using the TLS protocol, to obtain the necessary information that will be used to generate the current key used by all mesh routers to encrypt all the traffic transmitted on the mesh backbone. In particular, the device can connect to the wireless backbone in a secure way and begin executing the routing and access functions (phase 3).

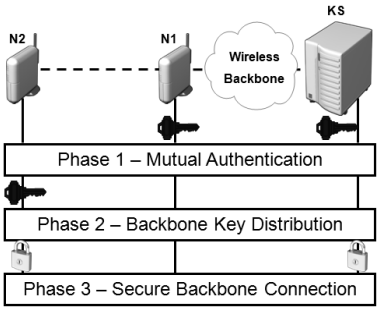


Fig. 1. MobiSEC security architecture: phases of the connection process performed by a new mesh router (node N_2). The depicted keys are used to encrypt backbone traffic.

During phase 2, mesh routers also perform a second authentication, based on the TLS protocol. Only authorized mesh routers that have the necessary credentials can authenticate to the Key Server and obtain the cryptographic material needed to derive the key sequence used to protect the wireless backbone.

In order to minimize the risks of using the same key for a long time and improve the overall security, each mesh router requests proactively the new cryptographic material (denoted as *session secret*) to the Key Server. According to the protocol used by all mesh routers, the *session secret* represents a key list (Server Driven Protocol) or a seed (Client Driven Protocol) that is used to compute the sequence of keys with a scheme that resembles a hash-chain method.

Figure 2 shows in detail the message exchanges that occur between the mesh router and the Key Server during the execution of both protocols.

A generic mesh router, after entering in the radio range of a mesh router already connected to the wireless backbone, authenticates itself to a central server and sends its first request to obtain the secret S used in the current session by the other routers that form the wireless backbone, and the time when it was generated, t_v, which represents the starting validity time. Let us define a $session$ as the maximum validity time of the secret currently shared by all nodes.

The node, based on the instant at which it joins the backbone (t_n in Figure 2), can compute an identifier of the key currently used by its peers (k_{id}), and its remaining validity time (T_1):

$$\begin{aligned} k_{id} &= \left\lfloor \frac{t_n - t_v}{timeout} \right\rfloor + 1 \\ T_1 &= key_{id} \cdot timeout - (t_n - t_v) \end{aligned} \tag{1}$$

According to the distribution protocol, the key identifier is used by the node to extract the current key from the provided session secret (Server Driven Protocol), i.e. $key(k_{id}, S) = S[k_{id}]$, or it represents the number of times the node has to apply a hash function to the session secret (Client Driven Protocol).

$$\begin{cases} key(k_{id}, S) = hash(S) & \text{if } k_{id} = 1 \\ key(k_{id}, S) = hash(key(k_i d - 1, S)) & \text{if } k_{id} > 1 \end{cases} \tag{2}$$

It is important that each node obtains the secret that will be used in the next session before the current session expires. In fact, if the request is sent when the current session is about to expire, the first nodes that receive the response will cut off the others when they enable the new key.

The key identifier that triggers the proactive request to the server is set equal to the difference between the maximum number of keys related to a specific session and a correction factor (c), which is computed based on the time necessary to receive the response from the Key Server (Δt), according to Equation (3). In this equation, t_s is the time when the first or proactive key request was sent, and t_r is the time when the corresponding key response was received from the Key Server. Hence, if a node takes a time (Δt in Equation (3)) greater than $timeout$ to receive the response from the Key Server, it must perform the next proactive request before setting the last key.

$$\begin{aligned} &\Delta t = t_r - t_s \\ &\begin{cases} c = \lceil \frac{\Delta t - timeout}{timeout} \rceil & \text{if } \Delta t \geq timeout \\ c = 0 & \text{if } \Delta t < timeout \end{cases} \end{aligned} \tag{3}$$

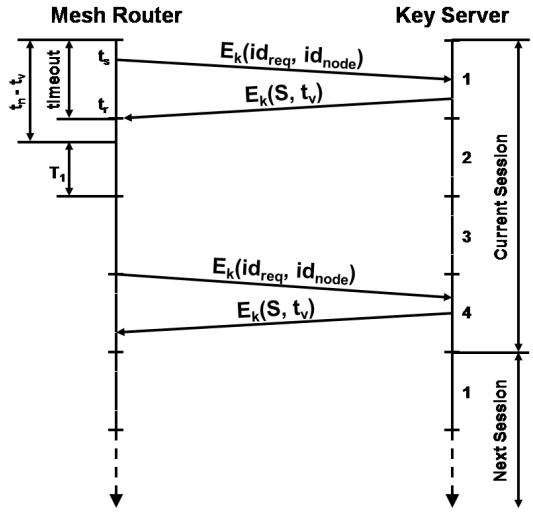


Fig. 2. *Key Distribution Protocol.* $E_k(\bullet)$ *represents the symmetric cryptographic function used to protect the security of the messages, whereas* id_{req} *and* id_{node} *represent the identifier of the request and of the node, respectively.*

III. Experimental Results

In this Section we present the numerical results obtained testing the proposed security framework within the MobiMESH test-bed, whose fundamental node is shown in Figure 3. The node, which can act both as router and access point, is an embedded system based on a VIA Epia Board equipped with a PCI-to-MiniPCI expander that permits to install four MiniPCI wireless cards.

We first measured the throughput of a long-lived TCP connection, whose segments are routed on a multi hop wireless path protected either by the Server Driven or the Client Driven

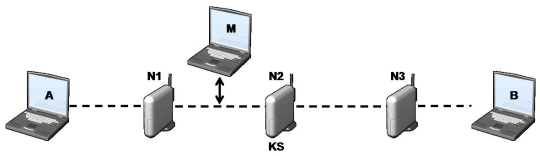
Fig. 3. Multi-radio MobiMESH router.

Protocol; then, we compared the obtained results with those achieved using a static key and an end-to-end approach like IPSec.

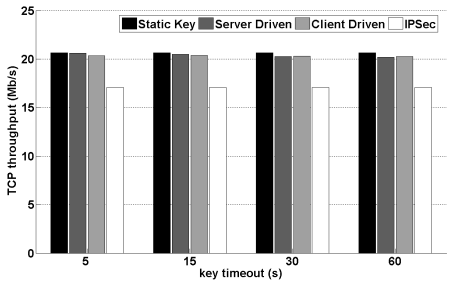
To prove the robustness of MobiSEC, we used a weak cryptographic system, i.e. WEP with a key length of 128 bit, and we tried to crack the key from the packets sniffed with the aircrack-ng tool, which implements the attack designed by Fluhrer, Mantin and Shamir (FMS attack) [7] with the KoreK improvements [8]. In all the tests we set the key timeout to 60 seconds and the session duration to 240 seconds. Such value is obtained by setting the maximum number of keys related to a specific session to 4.

We considered the multi-hop network scenario illustrated in Figure 4(a), where each router (N_1, N_2 and N_3) is connected only with the previous and the successive node.

Experimental results have been obtained generating TCP traffic between mesh clients A and B with the *iperf* traffic generator. All mesh routers run the client side application of the MobiSEC framework. Router N_2 also acted as Key Server, to evaluate the effect of the network load on our protocols, since in this configuration both N_1 and N_3 send their requests to N_2. Figure 4(b) shows the TCP throughput of the connection established between mesh clients A and B, secured by the considered protection schemes, as a function of the test duration. The results confirm that our solution, unlike IPSec, does not reduce the network performance with respect to a static key approach.



(a) Topology



(b) TCP throughput

Fig. 4. Multi-Hop Topology. A multi-hop data transfer between nodes A and C is performed to measure the network performance.

Strength analysis has been carried out in the same network scenario to evaluate how much our solution increases the overall security. Such analysis was performed sniffing the traffic transmitted between nodes N_1 and N_2 and then applying a cryptanalytic attack with the aircrack-ng tool. Table I reports the outcome of the cryptanalytic attack as a function of the time spent to gather the packets on which the attack is performed: only the static WEP key was broken, but the number of packets needed to get the key was significantly larger than the theoretical number indicated in [8]. In this works, the authors suggest that the number of packets necessary to crack a 128 bit WEP key is approximately $5 \cdot 10^5$ - 10^6, that is equivalent to 110-220 seconds considering an Ethernet packet and the theoretical throughput of an 802.11a/g wireless link. Therefore, setting the maximum key validity time to 60 s, as we did in MobiSEC, turns out to be quite a conservative choice.

TABLE I
OUTCOME OF THE CRYPTANALYTIC ATTACK.

	Packet-Gathering Time (s)		
Protocol	60	240	600
Static Key	Failed	Failed	Cracked (50 s)
Server Driven	Failed	Failed	Failed
Client Driven	Failed	Failed	Failed

Demonstration

We use three multi radio mesh nodes like those illustrated in Figure 3 and three laptops. Each node has up to four 802.11 a/g wireless interfaces. The demonstration aims to show the strength analysis described above as well as the throughput achievable by MobiSEC, using the topology illustrated in Figure 4(a).

In the first step, laptop A and B exchange data traffic and a video stream while the adversary laptop M performs a cryptanalytic attack on the traffic sniffed on the link N_1-N_2, which is protected by a static key. After breaking the key, M will show the video stream exchanged by the other two laptops.

In the second step, all mesh routers start executing the key distribution protocols in order to show the validity of the MobiSEC architecture, in terms of improved security and network performance.

ACKNOWLEDGMENTS

This work was partially supported by MIUR in the framework of the PRIN SESAME project.

REFERENCES

[1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, 2005.

[2] N. Ben Salem and J.-P. Hubaux. Securing wireless mesh networks. *IEEE Wireless Communications*, 13(2):50–55, 2006.

[3] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-securing ad hoc wireless networks. *ISCC '02*, pages 567–574, 2002.

[4] N. Milanovic, M. Malek, A. Davidson, and V. Milutinovic. Routing and security in mobile ad hoc networks. *IEEE Computer*, 2004.

[5] F. Martignon, S. Paris, and A. Capone. Design and Implementation of MobiSEC: a Complete Security Architecture for Wireless Mesh Networks *Computer Networks, article in press*, April 2009.

[6] A. Capone, S. Napoli, and A. Pollastro. MobiMESH: An experimental platform for wireless mesh networks with mobility supports. *WiMESH-Nets '06*. ACM, 2006.

[7] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. *Lecture Notes in Computer Science*, 2259:1–24, 2001.

[8] W. A. Arbaugh, N. Shankar, Y. C. J. Wan, and K. Zhang. Your 802.11 wireless network has no clothes. *IEEE Wireless Communications*, 9(6):44–51, 2002.