MobiSEC: A Novel Security Architecture for Wireless Mesh Networks

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

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

Antonio Capone
Department of Electronics and Information
Politecnico di Milano
capone@elet.polimi.it

ABSTRACT

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

In this paper we propose MobiSEC, a complete security architecture that provides both access control for mesh users and routers as well as security and data confidentiality of all communications that occur in the WMN.

MobiSEC extends the IEEE 802.11i standard exploiting the routing capabilities of mesh routers; after connecting to the access network as generic wireless clients, new mesh routers authenticate to a central server and obtain a temporary key that is used both to prove their credentials to neighbor nodes and to encrypt all the traffic transmitted on the wireless backbone links.

A key feature in the design of MobiSEC is its independence from the underlying wireless technology used by network nodes to form the backbone; furthermore, MobiSEC permits seamless mobility of both mesh clients and routers.

We implemented MobiSEC in a real-life test-bed and measured its performance in different network scenarios.

Numerical results show that our proposed architecture increases considerably the WMN security with a negligible impact on the network performance, thus representing an effective solution for wireless mesh networking.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

General Terms

Security, Performance

Keywords

Wireless Mesh Networks, Authentication, Security, Experimental Test-bed

Q2SWinet'08, October 27–28, 2008, Vancouver, BC, Canada.

1. INTRODUCTION

Wireless mesh networks (WMNs) have emerged recently as a technology for next-generation wireless networking [7, 11]. WMNs are the ideal solution to provide both indoor and outdoor broadband wireless connectivity in several environments without the need for costly wired network infrastructures.

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. Mesh routers are usually fixed and do not have energy constraints. WMNs, like wired networks, are characterized by infrequent topology changes and rare node failures.

Security in WMNs is still in its infancy as very little attention has been devoted so far to this topic by the research community [6, 7, 9]. Although many security schemes have been proposed for wireless LANs [27] and ad hoc networks [5, 18, 19, 20, 25, 30], they are not suitable for WMNs, which need convincing security solutions that should act as incentives for customers to subscribe to reliable services [7, 11, 13, 14].

In WMNs, two different security areas can be identified: one related to the *access* of users terminals (user authentication and data encryption), and the other related to network devices in the *backbone* of the WMN (mutual authentication of network devices and secure exchange of data and control messages).

In this paper we propose MobiSEC, a novel security architecture for wireless mesh networks that provides a complete security framework for both the access and backbone areas of the WMN, that is, access control for end-users and mesh routers as well as security and integrity of all data communications that occur in the WMN.

MobiSEC extends the IEEE 802.11i [1] standard to the WMN scenario exploiting the routing capabilities of wireless mesh routers. A two-steps 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.

We propose two key delivering protocols tailored for WMNs, named Server and Client Driven. In the Server Driven protocol all mesh routers require periodically to a central server (the Key Server) a new key list, whereas in the Client Driven protocol the mesh routers obtain from the server a seed and a hash function type to generate the cryptographic keys with a scheme similar to the hash chain method. Both protocols require a mutual authentication based on certificate exchanges between the mesh router and the server.

A key feature in the design of MobiSEC is its independence from the underlying wireless technology used by network nodes to form the backbone. Furthermore, MobiSEC allows seamless mobility of both mesh clients and routers. Client mobility is allowed by the 802.11i implementation, to which our solution is compliant, whereas mesh routers can roam freely around the backbone network after getting the key material from the Key Server, since all other mesh routers create the temporary key using the same information.

The proposed solution has been implemented and integrated in MobiMESH [12], 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 the numerical results show that our proposed scheme increases considerably the wireless mesh network security with a negligible impact on the network performance, thus representing an effective solution for wireless mesh networking.

The main contributions of this paper can therefore be summarized as follows:

- the proposition of MobiSEC, a novel security architecture for both the access and backbone areas of a WMN;
- the integration of the proposed solution in the experimental platform MobiMESH;
- a thorough evaluation with a test-bed of the proposed architecture in several realistic network scenarios.

The paper is structured as follows: Section 2 discusses related work. Sections 3 and 4 describe the proposed security framework and the key delivery protocols, while Section 5 provides an overview of the MobiMESH experimental platform.Section 6 discusses numerical results that show the effectiveness of our solution in various network scenarios. Finally, conclusions and directions for future research are presented in Section 7.

2. RELATED WORK

So far little attention has been devoted to security in WMNs by the research community [7, 9]. Two main security areas can be identified: the first is related to the access of client terminals, while the second is related to the mesh backbone.

Client authentication and access control can be provided using standard techniques [1, 2, 21], which guarantee a high level of flexibility and transparency: all users can access the mesh network without any change to their client device and software. However, client mobility can pose severe problems to security architectures, especially when real-time traffic is transmitted. To cope with these problems, proactive key distribution techniques can be devised [14, 17, 24].

Backbone security is another important issue. Mesh networks typically employ low-cost devices that cannot be protected against removal, tampering, or replication. If the device can be remotely managed, the adversary does not even need to physically access the router: a distant hacking into the device would work perfectly [9].

Several works investigate the use of cryptography techniques to secure the information exchanged through a wireless network. In [13] the authors propose to use PANA, the *Protocol for carrying Authentication for Network Access*, to authenticate the wireless clients and to provide them the cryptographic material necessary to create an encrypted tunnel with the remote access router to which they are associated. Even though such framework protects the confidentiality of the information exchanged over the network, it does not prevent adversaries to perform active attacks against the network itself. For instance, the topology information exchanged among mesh devices can be replicated, modified or forged, in order to deny access to users, steal the identity of legitimate nodes or assume sensible positions inside the network. Some preliminary solutions have been proposed in the sensor and ad hoc network research fields to prevent such attacks [5, 16, 23, 25, 26].

None of the above solutions, however, tackles all the security problems typical of a wireless mesh network. In fact, the previous proposals deal with security weaknesses related to a specific layer or protocol of the network stack. In this paper we propose a complete framework that copes with the security problems of both the access and backbone areas of a WMN, maintaining a high level of compatibility with current standards of wireless security without impacting on the WMN performance.

3. MOBISEC: A WIRELESS MESH NETWORK SECURITY ARCHITECTURE

In this Section we describe in detail the proposed solutions to provide both client and backbone security in a wireless mesh network.

Client security is guaranteed using the standard 802.11i protocol, while backbone security is provided with a two-steps approach: 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.

MobiSEC is independent from the underlying stream cipher technique adopted. In the numerical evaluation, however, we used WEP [1] to prove the robustness of the proposed solution even in the presence of a weak cryptographic system. We are currently implementing the CCMP algorithm for the IBSS operating mode [1], which is used by several mesh implementations (including MobiMESH) to establish the backbone links and form a multi-hop wireless architecture.

3.1 Assumptions

To specify the WMN scenario, we adopt the following definitions and assumptions:

- All nodes authorized to join the wireless backbone have two certificates that prove their identity: one is used during the authentication phase that occurs when a new node joins the network (we use EAP-TLS [4] for the 802.1X authentication, since it is the strongest authentication method provided by EAP [3, 28]), whereas the second certificate is used for the mutual authentication with the Key Server.
- The certificate used during the mutual authentication with the RADIUS server and the one used during the mutual authentication with the Key Server are signed by the same Certification Authority (CA). Only recognized mesh routers are authorized to join the backbone, by providing them the necessary cryptographic material used by other devices to protect the wireless backbone.
- Synchronization of all mesh routers is needed; this can be obtained using for example the NTP protocol.

3.2 Client Security

To achieve the highest possible level of transparency, the access mechanism to the wireless mesh network is designed to be identical to that of a generic wireless LAN, where mobile devices connect to an access point. Since almost each wireless device currently available on the market implements the security functionalities described in the IEEE 802.11i protocol [1], we propose to configure mesh routers to comply with such standard. This solution allows users to access the mesh network exploiting the authentication and authorization mechanisms without installing additional software.

Evidently, such security solution protects only the wireless access link between end clients and access nodes. However, an adversary could eavesdrop the data exchanged on the wireless mesh network, unless a security system is implemented to protect the backbone links.

Figure 1 illustrates such a situation in which a data exchange occurs between Alice and Bob, who are connected in a secure way to wireless mesh routers 1 and 2, respectively (these nodes also act as WPA/WPA2 Access Points). If the wireless link established between such routers is not protected by any security system, Mallory will be able to eavesdrop the communication, since nodes 1 and 2 will forward the traffic on the wireless link on which Mallory is listening. This situation is prevented by MobiSEC, which encrypts all the traffic transmitted on the wireless link with a stream cipher operating at the data link layer.

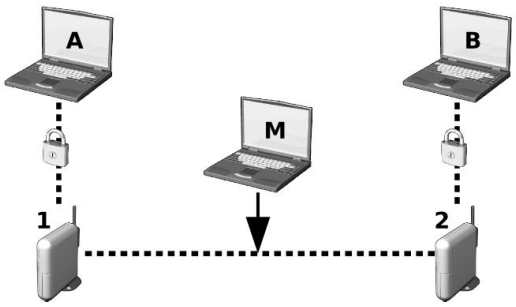

Figure 1: Alice and Bob exchange data through the wireless mesh network. Mallory will be able to eavesdrop their data, unless a security system is implemented to protect the backbone link.

3.3 Backbone Security

The client security solution illustrated above provides confidentiality and integrity of the information transmitted only on the wireless access link. Therefore, we propose an additional system to secure communications that occur over the wireless backbone. A two-steps approach is adopted, in which new nodes dynamically join the network as wireless clients and afterwards can upgrade their role becoming wireless mesh routers by further authenticating to a Key Server.

Two major problems arise: on one hand it is necessary to authenticate new mesh routers that join the network, and provide them the cryptographic material needed to derive keys that make secure data transfer possible. On the other hand, it is important to develop a system with a minimum impact on device mobility. To this aim, we designed and implemented a key delivering solution that exploits the existing access network, allowing a new node to connect to a remote server which sends the temporary key used by all mesh routers to encrypt the traffic transmitted over the wireless backbone. Such key represents the proof that the new node has the required credentials to become a mesh router.

Figure 2 shows the three phases of the connection process performed by a new mesh router (namely, node N). When N wants to connect to the mesh network, it scans all radio channels to detect a mesh router already connected to the wireless backbone, that is therefore able to provide access to all network services (including authentication and key delivering). Let A be such router. After connecting to A, N can perform the tasks described by the IEEE 802.11i 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 other words, during this phase N performs all the activities as a generic wireless client to establish a secure channel with a mesh router (node A in our example) that can forward its traffic securely over the wireless backbone.

In phase 2, node N connects to the Key Server (KS) 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. The generated key represents the proof that N has the necessary credentials to cooperate with other nodes to maintain the wireless backbone; in particular the device can connect to the wireless backbone in a secure way and start executing the routing and access functions (phase 3).

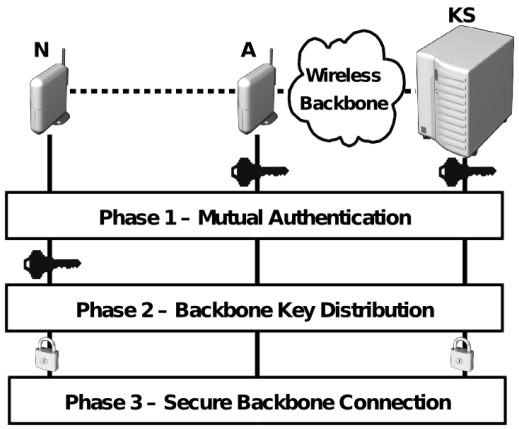

Figure 2: Phases of the connection process performed by a new mesh router (node N). The depicted keys are used to encrypt the backbone traffic.

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 our architecture, an end-to-end secure channel between the Key Server and the mesh router is established at the end of the successful authentication through which the cryptographic material can be exchanged in a secure way.

To minimize the risks of using the same key for a long interval, we propose two key delivery and regeneration protocols, described in Section 4, to create a new key when a predetermined timeout expires. Both protocols require the synchronization of all mesh routers with a central server.

Figure 3 shows an example network composed of 4 mesh routers connected with 5 wireless links, represented with dashed lines, and the Key Server (KS). Our proposed solution permits an automated and incremental configuration process of the wireless mesh network. At the beginning of the process, only node A can connect to

the mesh network, since it is the only node that can complete the authentication with the Key Server and get from it the cryptographic material needed to set up an ad hoc and protected wireless link. The neighbors of A (B and C) detect a wireless network to which they can connect and perform the authentication process described by the 802.11i standard as generic wireless clients. Through the wireless network, the mesh routers will be able to authenticate with the Key Server to request the information used by A to produce the currently used cryptographic key. After having derived such key, both B and C will be able to reach each other, as well as node A, in ad hoc mode. Moreover they will be able to turn on their access interface through which they will provide to node D a network connection towards the server.

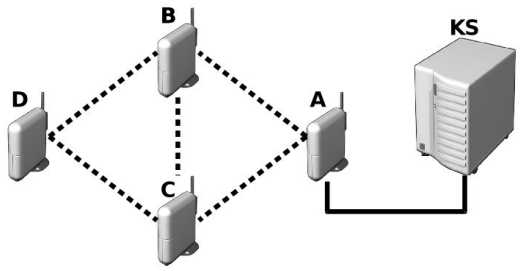

Figure 3: Example of the proposed automated WMN configuration process. Our proposed solution permits an automated and incremental configuration process of the wireless mesh network.

4. KEY DELIVERING PROTOCOLS

In this Section we describe two protocols, denominated *Server Driven* and *Client Driven*, that we propose to perform the key delivery and regeneration tasks. The main difference between such protocols is that the former grants the mesh router more autonomy than the latter during the key regeneration process.

4.1 Server Driven Protocol

This protocol provides a reactive method to deliver the keys used by all mesh routers to protect the integrity and confidentiality of the traffic exchanged during a specific interval. Figure 4 shows in detail the message exchanges that occur between the mesh router and the Key Server. A generic mesh router, after a successful mutual authentication with a central server, sends its first request to obtain the key list used in the current session by the other mesh routers that form the wireless backbone and the time when it was generated, Key List Timestamp (TS_{KL}). Let us define a $session$ as the maximum validity time of the key list currently used by each node; its duration is the product of the key list cardinality (i.e. the number of the keys in the list) and the maximum validity time of a generic key (the parameter $timeout$ in Figure 4). Moreover, the key list validity starts when it is generated, i.e. at TS_{KL}. The node, based on the instant in which it joins the backbone (t_{now} in Figure 4), can find out the key, among those in the list, currently used by its peers and its validity time (key_{idx} and T_1) according to the following expression:

$$\begin{aligned} key_{idx} &= \left\lfloor \frac{t_{now} - TS_{KL}}{timeout} \right\rfloor + 1 \\ T_1 &= key_{idx} \cdot timeout - (t_{now} - TS_{KL}) \end{aligned} \quad (1)$$

It is important that each node requests the server the key list that will be used in the next session before the current session expires. This is especially true for nodes that take a long time to receive the response from the server (due, for example, to slow links or high number of hops from the server). In fact, if the request is sent when the current session is almost to expire, the nodes that are connected to the server with the fastest links will receive the response before other nodes; hence they will cut off the others when they enable the new key.

The key index value that triggers the proactive request to the server can be set equal to the difference between the key list cardinality and a correction factor, that can be estimated based on parameters like network load, the distance to the server and the previous time to get the response.

In our architecture, the correction factor is based on the time necessary to receive the response from the Key Server and it is estimated according to Equation (2), where t_s is the time when the first or proactive key request was sent, t_r is the time when the key response was received from the Key Server and $timeout$ is the maximum key validity time. So, if a node takes a time (t_{last} in Equation (2)) greater than $timeout$ to receive the response from the Key Server, it must perform the next proactive request before setting the last key (otherwise, it will not have enough time to get the response).

$$\begin{aligned} &\begin{cases} c = \left\lceil \frac{t_{last} - timeout}{timeout} \right\rceil & \text{if } t_{last} \geq timeout \\ c = 0 & \text{if } t_{last} < timeout \end{cases} \\ &t_{last} = t_r - t_s \end{aligned} \quad (2)$$

To illustrate how the correction factor is evaluated, let us refer to the example message exchange shown in Figure 4; the router performs the second request when the third key is set (i.e. the correction factor is equal to 1), so it has enough time to receive the response from the Key Server. In this example, in fact, during the first message exchange it has taken a time greater than $timeout$ to get the response (i.e. t_{last}).

Note that the first request of the key list sent by the new mesh router to the Key Server will be forwarded by the peer to which it is connect as generic wireless client through the wireless access network, while successive requests will be sent directly over the wireless backbone.

Since in MobiSEC the connection with the server is reliable (the protocol used to exchange key material is SSL that is based on TCP), it is unnecessary to include in the architecture an acknowledgment mechanism.

4.2 Client Driven Protocol

The Client Driven protocol grants mesh routers more autonomy in the key regeneration process with respect to the Server Driven protocol. In fact, the server provides only a seed and a function type that must be used to compute the sequence of keys used by mesh nodes, with a scheme that resembles a hash-chain method. Figure 5 shows the message exchanges performed between the mesh router and the Key Server. As in the previous protocol, a generic mesh router, after a successful mutual authentication with a central server, sends its first request to obtain the seed currently used by the other backbone nodes to create the key sequence, and the time when it was generated, Seed Timestamp (TS_{seed}). Hence, in the Client Driven Protocol, a $session$ is defined as the validity time of the current seed and its duration is the product of the maximum number of keys generated with the same seed and the validity time of a generic key (the parameter $timeout$ in Figure 5). Equation (3) illustrates how to compute the value assumed by r, which indicates

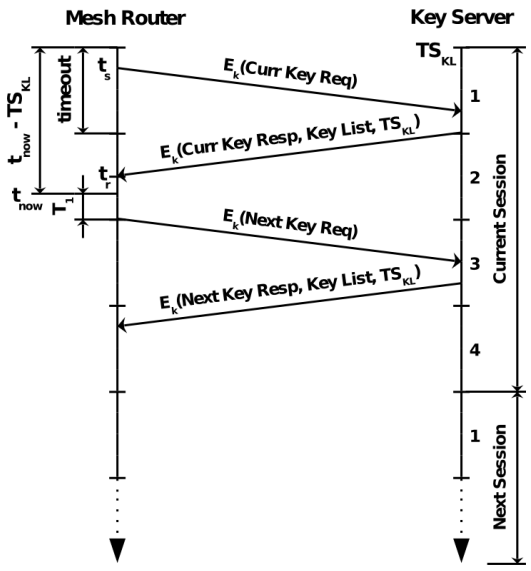

Figure 4: Server Driven Protocol. Since the first request has taken a time greater than $timeout$, the mesh router sends the successive request when it sets the third key.

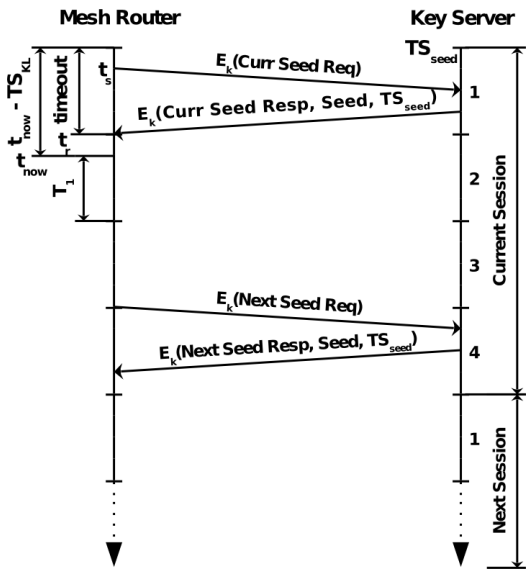

Figure 5: Client Driven Protocol. Since the first request has taken a time smaller than $timeout$, the mesh router sends the successive request when it sets the fourth key.

how many times the mesh router must apply the hash function to synchronize its first key with that currently used by the other nodes. The value assumed by T_1 specifies the validity time of the generated key.

$$\begin{cases} key(r, seed) = hash(seed) & \text{if } r = 1 \\ key(r, seed) = hash(key(r-1, seed)) & \text{if } r > 1 \end{cases}$$
$$r = \left\lfloor \frac{t_{now} - TS_{KL}}{timeout} \right\rfloor + 1 \quad (3)$$
$$T_1 = r \cdot timeout - (t_{now} - TS_{seed})$$

To enhance the security of the entire system the following features are added:

- the argument of the hash function can be obtained by concatenating the seed and the timestamp with a pre-shared secret known by each node;
- a maximum interval for the validity of the seed is set.

Even if an adversary obtains a key with crypto-analysis of the traffic exchanged over the network, he would have access to the mesh backbone only during the time that remains before the new key regeneration. The new seed can be obtained by all mesh routers with the same proactive mechanism described above for the Server Driven Protocol. Hence, when the mesh router generates one of the last keys that can be computed with the current seed (the one that allows the node to receive the response from the Key Server), it sends a request for a new seed to the server. In Figure 5 the router performs such proactive request when the fourth key is generated, since the time spent to get the seed response after sending the first request is less than the key timeout. In this case the correction factor is null, as the $timeout$ value is long enough to get the response before the session expiration.

As for the previous protocol, the first seed request is sent by the new mesh routers to the Key Server through the wireless access network, while successive requests will be sent directly over the wireless backbone.

Note that the proposed architecture can be easily applied to a multi-radio WMN, where each node is endowed with several wireless interfaces dedicated to the backbone traffic. To this end, it is necessary to modify simply the response format of the Key Server to distinguish the cryptographic information (key list or seed and type of the hash function) that is related to the different radio interfaces. A distinct process can be executed for each radio interface, specifying for each of them a different Key Server. Hence, MobiSEC can be extended without changing the source code. In the Numerical Results Section we report for simplicity the results obtained with a single backbone interface for each mesh router.

5. MOBIMESH ARCHITECTURE

In the following we provide a brief description of the MobiMESH architecture [12], the experimental platform on which we evaluated the performance of our solution.

MobiMESH is designed following the hybrid mesh network architecture paradigm. It is therefore composed by a mesh backbone core section, which is responsible for routing, mobility and security management, and by an access network, which hosts IEEE 802.11 WLAN clients. Figure 6 illustrates the architecture of the MobiMESH network.

The backbone network, where all devices perform the routing and security protocols to form and maintain a multi-hop wireless architecture, is based on the ad hoc network paradigm.

The access network is designed so that clients perceive the network as a standard IEEE 802.11 WLAN and behave accordingly; MobiMESH can therefore be accessed by standard WLAN clients without installing additional software.

The fundamental node of the MobiMESH backbone network is an integrated device that acts both as router and access point. Such device is equipped with at least two radio interfaces, one of which is used to establish the wireless links with the other mesh routers of the backbone network, while the other serves as access point for the access network.

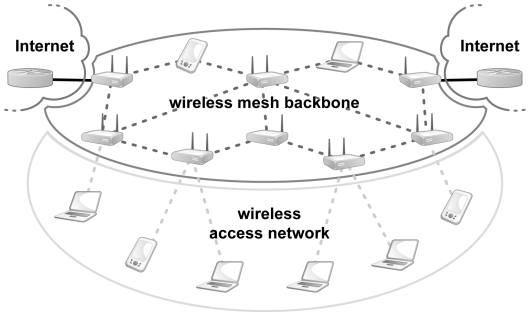

Figure 6: MobiMESH architecture.

Figure 7 shows a sample node on which we implemented the MobiSEC architecture. The node 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. The black external antenna provides access to the wireless clients, whereas the other antennas form the wireless backbone links with the other mesh routers.

Figure 7: Multi-radio MobiMESH router.

6. NUMERICAL RESULTS

In this Section we present the numerical results obtained testing the proposed security framework within the MobiMESH test-bed, considering different network scenarios.

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) [15] with the KoreK improvements [8, 22, 29]. 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 Key List cardinality (for the Server Driven Protocol) and the maximum number of keys created with the same seed (for the Client Driven Protocol) to 4.

6.1 Full-Mesh Topology

We first considered the full-mesh network topology illustrated in Figure 8, where each router is directly connected with the other two nodes (all nodes belong to the same ad hoc wireless cell). In such scenario we first measured the throughput of a long-lived TCP connection established over a wireless link protected either by the Server Driven or the Client Driven Protocol; then, we compared the obtained results with those achieved on a radio link protected with a static key. Numerical results have been obtained generating TCP traffic between mesh routers A and C with the D-ITG traffic generator [10]. Router C also acted as Key Server, to evaluate the effect of the network load on our protocols, since in this configuration both A and B send the key material request to C. Figure 9 shows the TCP throughput of the wireless link between nodes A and C secured by the three protection schemes considered as a function of the test duration. The average throughput was equal to 25.7 Mb/s for a link protected by a static key, 25.7 and 24.7 Mb/s for a link protected by a dynamic key (generated respectively by the Client Driven and the Server Driven Protocol). The results confirm that our solution does not reduce the performance of the wireless link.

At the same time, we tested the availability of the Key Server installed on node C, even in the presence of a high network load. We verified that all mesh routers could remain connected by checking the connection status of the wireless links established among the nodes.

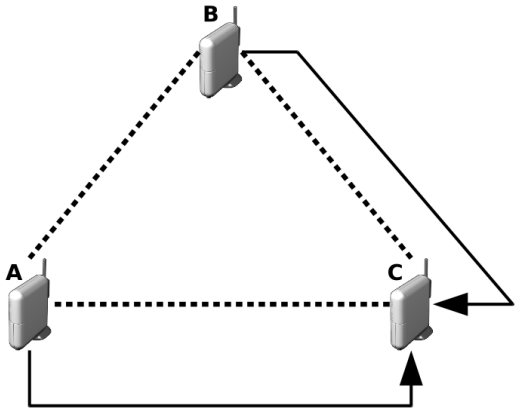

Figure 8: Full-Mesh Topology. A data transfer is performed between nodes A and C. Although C also acts as Key Server, the connection among the three nodes remain available.

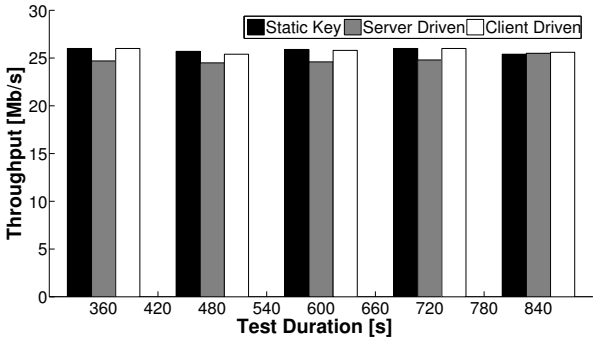

Figure 9: TCP throughput measured in the full-mesh network scenario for different key delivering protocols.

In the same scenario we further measured the packet loss eventually caused by the key renewal procedure, considering a data transfer based on a UDP connection. Packet loss can be critical for real-

time multimedia applications, such as VoIP and streaming video. We therefore generated UDP traffic on a wireless link, performing a data transfer between mesh routers B and C, and we compared the performance of our protocols with that obtained on the same wireless link secured by a static key. The transmission rate was set to 1 Mb/s and several data transfer sessions were performed, each with a duration ranging in the 2 to 12 minutes interval. Since we measured a negligible packet loss in all our experiments, we can affirm that our security solution has no impact on such performance figure, even when several key switchings are performed.

Strength Analysis

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 A and C and then applying a crypto-analytic attack with the aircrack-ng tool. Table 1 reports the outcome of the crypto-analytic 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, 22, 29]. In these 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. Increasing the fudge factor, which is related to the number of secret keys to try (i.e. the brute force of the attack) [22], had no effect on the results of attacks against our protocols: in both cases aircrack-ng failed to recover the keys used to encrypt the frames. The longer execution time took by the tool to crack the key in the third session was due to the greater numbers of keys that aircrack-ng tried.

Table 1: Full-Mesh Topology: Key Cracking Time. The $key\ timeout$ **and** $session\ duration$ **parameters were set to 60 s and 240 s, respectively. The** $packet\ gathering\ time$ **varied from 60 to 600 seconds.**

Fudge Factor = 2

	Packet-Gathering Time (s)		
Protocol	60	240	600
Static Key	Failed	Failed	Cracked (5 s)
Server Driven	Failed	Failed	Failed
Client Driven	Failed	Failed	Failed

Fudge Factor = 4

	Packet-Gathering Time (s)		
Protocol	60	240	600
Static Key	Failed	Failed	Cracked (7 s)
Server Driven	Failed	Failed	Failed
Client Driven	Failed	Failed	Failed

6.2 Multi-hop Topology

We then considered the multi-hop network scenario illustrated in Figure 10. We performed a UDP data transfer between nodes A and D using the D-ITG traffic generator, and we measured the performance of the proposed protocols. All mesh routers run the client side application of the Client Driven Protocol, and node D also acted as Key Server. The total time of the test was 480 s, the packet rate 1 Mb/s and the packet size was set to 1500 byte. Also in this scenario we observed that the packet loss was negligible. Table 2 shows the network performance measured by D-ITG. The packet delivery delay is practically constant (the value of its standard deviation is very low) which guarantees a correct operation even for real-time multimedia applications. Moreover, the low mean jitter value suggests that our solution introduces no perceptible alterations in the transmitted voice stream.

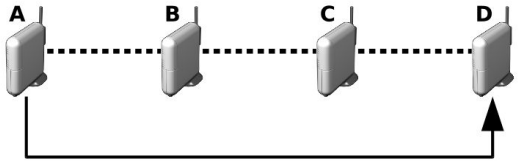

Figure 10: Multi-Hop Topology. A multi-hop data transfer between nodes A and D is performed to measure the network performance.

Table 2: *Multi-hop Performance*

Parameter	Value (ms)
Delivery Delay Mean	3.1
Minimum Delivery Delay	2.7
Maximum Delivery Delay	5.8
Delivery Delay Std Deviation	0.8
Mean Jitter	0.8

We also evaluated the MobiSEC framework in a scenario composed of 10 randomly placed nodes. For the sake of brevity we do not report the obtained results, since they confirm the network performance illustrated above.

7. CONCLUSION

In this paper we proposed MobiSEC, a novel security architecture tailored for wireless mesh networks. MobiSEC tackles the security problems of both the access and backbone areas of WMNs, providing an effective and transparent security solution for end-users and mesh nodes.

We implemented our proposed security architecture in MobiMESH, a complete wireless mesh network framework, and we tested it in several realistic network scenarios.

Numerical results show that MobiSEC offers secure network services to both mesh users and routers with negligible impact on network performance (in particular, on the transmission rate and network latency), therefore representing an effective solution for wireless mesh networking.

Future research issues include the study of a distributed and collaborative system where the authentication service is provided by a dynamically selected set of mesh routers. The integration with the current centralized scheme would increase the robustness of our solution, maintaining a low overhead since mesh routers would use the distributed service only when the central server is not available.

Acknowledgments

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

8. REFERENCES

[1] *IEEE Standard 802.11i. Medium Access Control (MAC) security enhancements, amendment 6*. IEEE Computer Society, 2004.

[2] *IEEE Standard 802.1X. Port-Based Network Access Control*. IEEE Computer Society, 2004.

[3] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz. Extensible Authentication Protocol (EAP). *RFC 3748*, June 2005.

[4] B. Aboba and D. Simon. PPP EAP TLS authentication protocol. *RFC 2716*, October 1999.

[5] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Mühlethaler, and D. Raffo. Securing the OLSR protocol. In *Proceedings of the IFIP Med-Hoc-Net, Mahdia, Tunisie*, June 2003.

[6] C. Adjih, D. Raffo, and P. Mühlethaler. Attacks against OLSR: Distributed key management for security. In *Proceedings of the 1st OLSR Interop and Workshop*, August 2005.

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

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

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

[10] A. Botta, A. Dainotti, and A. Pescapé. Multi-protocol and multi-platform traffic generation and measurement. In *Infocom '07 DEMO Session*, volume 45, pages 526–532, May 2007.

[11] R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. *Communications Magazine, IEEE*, 43(3):123–131, March 2005.

[12] A. Capone, S. Napoli, and A. Pollastro. MobiMESH: An experimental platform for wireless mesh networks with mobility supports. In *WiMESHNets '06: Proceedings of the 1st ACM workshop on Wireless mesh: moving towards applications*. ACM, August 2006.

[13] O. Cheikhrouhou, M. Laurent-Maknavicius, and H. Chaouchi. Security architecture in a multi-hop mesh network. June 2006.

[14] R. Fantacci, L. Maccari, T. Pecorella, and F. Frosali. A secure and performant token-based authentication for infrastructure and mesh 802.1X networks. April 2006.

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

[16] Y.-C. Hu, A. Perrig, and D. B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Springer*, 11(1-2):21–38, January-February 2005.

[17] M. Kassab, A. Belghith, J.-M. Bonnin, and S. Sassi. Fast pre-authentication based on proactive key distribution for 802.11 infrastructure networks. In *WMuNeP '05: Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modeling*, pages 46–53. ACM, 2005.

[18] N. Komninos, D. Vergados, and C. Douligeris. Detecting unauthorized and compromised nodes in mobile ad hoc networks. *Ad Hoc Netw.*, 5(3):289–298, 2007.

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

[20] N. Milanovic, M. Malek, A. Davidson, and V. Milutinovic. Routing and security in mobile ad hoc networks. *Computer*, 37(2):61–65, February 2004.

[21] A. Mishra and W. A. Arbaugh. An initial security analysis of the IEEE 802.1X standard. *UM Computer Science Department, Technical Report CS-TR-4328*, February 2002.

[22] M. Ossmann. WEP: Dead again. *http://securityfocus.com/infocus/1814*, 2004.

[23] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, May 2005.

[24] A. Prasad and H. Wang. Roaming key based fast handover in WLANs. *Wireless Communications and Networking Conference, 2005 IEEE*, 3:1570–1576, March 2005.

[25] D. Raffo, C. Adjih, T. Clausen, and P. Mühlethaler. An advanced signature system for OLSR. In *SASN '04: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 10–16, 2004.

[26] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer. A secure routing protocol for ad hoc networks. *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, pages 78–87, November 2002.

[27] W. Stallings. *Cryptography and Network Security, Fourth Edition*. McGraw-Hill, September 2003.

[28] D. Stanley, J. Walker, and B. Aboba. Extensible Authentication Protocol (EAP) method requirements for wireless LANs. *RFC 4017*, March 2005.

[29] A. Stubblefield, J. Ioannidis, and A. D. Rubin. A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP). *ACM Trans. Inf. Syst. Secur.*, 7(2):319–332, 2004.

[30] L. Zhou and Z. Haas. Securing ad hoc networks. *Network, IEEE*, 13(6):24–30, November 1999.