Computer Networks 53 (2009) 2192–2207

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Design and implementation of MobiSEC: A complete security architecture for wireless mesh networks

Fabio Martignon [a], Stefano Paris [b], Antonio Capone [b,*]

[a] Department of Information Technology and Mathematical Methods, University of Bergamo, Italy
[b] Department of Electronics and Information, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

ARTICLE INFO

Article history:
Received 23 October 2008
Received in revised form 18 March 2009
Accepted 6 April 2009
Available online 17 April 2009

Responsible Editor: T. Melodia

Keywords:
Wireless mesh networks
Authentication
Security
Experimental testbed

ABSTRACT

Wireless mesh networks (WMNs) have emerged recently as a technology for next-generation wireless networking. They consist of mesh routers and clients, where mesh routers are almost static and form the backbone of WMNs. WMNs provide network access for both mesh and conventional clients.

In this paper we propose MobiSEC, a complete security architecture that provides both access control for mesh users and routers as well as a key distribution scheme that supports layer-2 encryption to ensure 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 allows seamless mobility of both mesh clients and routers.

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. Moreover, extensive simulations have been performed in large-scale network scenarios using Network Simulator.

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

1. Introduction

Wireless mesh networks (WMNs) have emerged recently as a technology for next-generation wireless networking [1,2]. 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 [1,3,4]. Although many security schemes have been proposed for wireless LANs [5] and ad hoc networks [6–11], they are not suitable for WMNs, which

* Corresponding author. Tel.: +39 02 2399 3449; fax: +39 02 2399 3413.
E-mail addresses: fabio.martignon@unibg.it (F. Martignon), paris@elet.polimi.it (S. Paris), capone@elet.polimi.it (A. Capone).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.04.002

need convincing security solutions that should act as incentives for customers to subscribe to reliable services [1,2,12,13].

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. 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 [14] 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.

We propose two key distribution protocols tailored for WMNs, named Server and Client Driven. In the Server Driven protocol, all mesh routers periodically send a request to a central server (the Key Server) to obtain 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 [15], 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. Furthermore, we extended the Network Simulator (ns v.2) [16] implementing the MobiSEC architecture, and performing extensive simulations in large-scale network scenarios to test the behavior of our architecture also in the presence of a large number of nodes and traffic flows.

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. The first approach provides an upper bound in terms of achievable throughput, delay and packet losses, while the latter is useful to gauge the performance gap between our proposed architecture and existing end-to-end security solutions. Numerical results show that MobiSEC considerably increases 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 complete 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 of the proposed architecture in several realistic network scenarios, using both a testbed and simulations.

The paper is structured as follows: Section 2 discusses related work. Section 3 introduces the network and threat models considered in our work. Section 4 provides an overview of the MobiMESH experimental platform, as well as of the 802.1X and 802.11i standards. Sections 5 and 6 describe the proposed security framework and the key distribution protocols, respectively. Section 7 discusses numerical results that show the effectiveness of our solution in several realistic network scenarios. Finally, conclusions are presented in Section 8.

2. Related work

So far, little attention has been devoted to security in WMNs by the research community [1,3]. 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 [14,17,18], which guarantee a high level of flexibility and transparency: all users can access the mesh network without any change to their client devices 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 [13,19,20].

Several works investigate the use of cryptographic techniques to secure the information exchanged through a wireless network. In [12] the authors propose to use PANA, the *Protocol for carrying Authentication for Network Access*, to authenticate the wireless clients and to provide them with the cryptographic material necessary to establish an encrypted tunnel with the remote access router to which they are associated.

Other approaches have been proposed to authenticate the users in WMNs, maintaining at the same time a low overhead. In [21] a security architecture for high integrity multi-hop WMNs is proposed; a heterogeneous set of

WMN providers is modeled as a credit-card based system so that each mesh client does not need to be bound to a specific operator, but can achieve ubiquitous network access by first obtaining a universal pass issued by a trusted third broker. Such an approach is suitable for WMNs managed by multiple operators, whereas in this paper we are interested in a scenario where a single operator manages the WMN and is liable for all the authentication procedures.

The authors of [22] define a new authentication technique for hierarchical WMNs based on threshold cryptography, where the certification authority services are provided through the collaboration of a pre-determined set of mesh routers. The proposed architecture extends the Diffie–Hellman key exchange protocol for negotiating a key that authorizes a user to access the backbone network services provided by a mesh router situated in a different zone.

Even though such frameworks protect the confidentiality of the client information exchanged over the network, they do not prevent adversaries from performing 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.

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 [3].

Some preliminary solutions have been proposed in the sensor and ad hoc network research fields to detect and prevent such attacks. In [8] the authors propose a distributed detection mechanism that makes use of local agents to collect and analyze audit data. Each agent assigns a compromised status based on its data analysis and passes it to the neighboring nodes for further decisions. In [23], two protocols are defined to detect replicated nodes by distributing the information about own identity and position (e.g. geographical coordinates) to a randomly selected set of nodes. The Birthday Paradox guarantees that in a high density network both protocols can detect an identity collision with high probability.

Similarly to the work in [24], where a suite of security protocols optimized for Wireless Sensor Networks is proposed, our architecture is based on a central entity that is liable for the authentication and key management services. The hardware constraints of sensor nodes, however, force the authors of [24] to define authentication methods based only on symmetric techniques, where the base station needs to maintain a symmetric key with each sensor node. On the other hand, in MobiSEC we use asymmetric cryptographic functions to authenticate the network nodes, which require only the knowledge of the public key of the certification authority that released all the certificates.

Other works investigate the use of threshold cryptography to achieve high fault tolerance against network partitioning. The work presented in [9] defines two different approaches to allow specific coalitions of devices to act together as a single certificate authority, whereas in [25] a hierarchical key management architecture is proposed to obtain an efficient establishment of distributed trust.

Even if these distributed systems improve the network fault tolerance by removing the single point of failure introduced by centralized schemes, they are not very efficient in terms of computational or communication overhead.

Different attacks against signaling and routing protocols are analyzed in [4,6,26,27], where the authors propose some modifications to mitigate these attacks.

None of the above solutions, however, addresses 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, while 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. System models and assumptions

In order to specify the WMN scenario we are dealing with, we present the communication and threat models considered in our architecture, as well as the definitions and assumptions we adopt in the design of MobiSEC.

3.1. Network model

This work considers an hybrid wireless mesh network, where all network devices communicate with each other using the wireless medium, which is intrinsically insecure due to its broadcast nature. We assume that all wireless links established between any two nodes are symmetric, and we do not consider in this paper security issues deriving from asymmetric channels.

In this type of WMNs, mesh routers form the backbone network by collaborating in the execution of management and control operations, whereas mesh clients can access all network services, including authentication and key management, through mesh routers. We further assume that each mesh router is endowed with at least two wireless interfaces: one is used to provide user access, while the remaining interfaces are used for backbone communications.

Since the WMN architecture we consider has a hierarchic structure (wireless mesh routers are in fact dedicated nodes which are deployed to offer backhaul services), we assume the existence of a network operator that is liable for all managements tasks.

Finally, as we explain in the following sections, the security procedures defined in MobiSEC require the synchronization of all mesh routers. This can be achieved using for example the NTP protocol.

3.2. Adversary models and security assumptions

The broadcast nature of the wireless medium makes WMNs prone to security attacks, which can be classified into two categories: *passive* attacks, like eavesdropping, where malicious users violate the confidentiality of the

information exchanged over the network, and *active* attacks, which involve actions performed by adversaries to gain the control of the network. For example, the topology and signaling information exchanged between network devices can be replicated, modified or forged, in order to deny access to users, steal the identity of legitimate nodes or get sensible positions inside the network.

In the design of MobiSEC, we consider also adversaries that try to gain network access by performing cryptanalytic attacks on the exchanged traffic to recover the encryption keys and elude the authentication and authorization services.

In our proposed architecture, the security of the network infrastructure is obtained using standard encryption techniques that permit to achieve both confidentiality and integrity of the exchanged traffic. In particular, we adopt a hop-by-hop encryption scheme based on the cryptographic functions operating at the data link layer, in order to provide a unique solution to secure both data and signaling communications against external attacks. Such a solution requires the design of an authentication method to verify the identity of network devices, and a key distribution service to deliver the necessary cryptographic information according to the result of the authentication procedure and the node's role.

To distinguish user terminals from nodes authorized to join the wireless backbone, these latter devices have two certificates that prove their identity: one is used during the authentication phase that occurs when a new node joins the access network, whereas the second certificate is used for the mutual authentication to gain backbone access. Only recognized mesh routers are authorized to join the backbone, and the Key Server provides them with the necessary cryptographic material used by all network devices to protect the wireless backbone. This solution allows the network operator to set up a public key infrastructure (PKI) for backbone nodes independently from users' PKI, and potentially, it could allow to separate the infrastructure management from the access service.

Finally, we assume the existence of at least one tamper resistant mesh router that hosts the authentication and key management services described in the following sections. Since this special node cannot be compromised, it is considered trustworthy by all the other network nodes, in the sense that it is assumed to behave correctly.

4. Overview of the MobiMESH architecture and of the 802.11i protocol

In this section we first provide an overview of the MobiMESH architecture [15], the experimental platform on which we implemented and evaluated the proposed solution. Then we review briefly the IEEE 802.11i standard, starting from the 802.1X protocol that defines the procedures to authenticate a new user wanting to join the access network, since it is part of MobiSEC.

4.1. MobiMESH architecture

Fig. 1 illustrates the architecture of the MobiMESH network, which is designed following the hybrid mesh network architecture paradigm. It is therefore composed of 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.

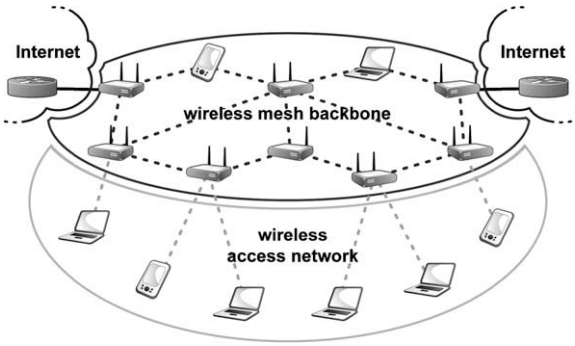

Fig. 1. MobiMESH architecture.

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.

Fig. 2 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 the installation of 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.

Fig. 2. Multi-radio MobiMESH router.

4.2. The IEEE 802.1X port based access protocol

The IEEE 802.1X standard provides a general architecture to authenticate devices and to authorize the network access. The protocol defines three entities that participate in the authentication process:

- the Supplicant, which represents a new device that requires access to the network and must prove its identity;
- the Authenticator, which is the device placed at the end of a point-to-point connection that allows the Supplicant to connect to the network;
- the Authentication Server, which provides the authentication service to the Authenticator, i.e. it verifies the credentials provided by the Supplicant and sends to the Authenticator the authorization status of the new device. The Authentication Server must also create the cryptographic material that is used by the other entities to derive the session keys.

The Supplicant and the Authentication Server use the Extensible Authentication Protocol (EAP) [28] to exchange their credentials and prove their identities to each other. During this phase, the Authenticator forwards the messages sent by the Supplicant to the Authentication Server and it delivers to the Supplicant the responses from the Authentication Server. In particular, the Supplicant encapsulates the EAP messages in EAPOL frames (Extensible Authentication Protocol Over LAN) [29]. When the Authenticator receives an EAPOL frame, it removes the header and it sends the EAP message to the Authentication Server using the RADIUS protocol (Remote Authentication Dial In User Service) [30]. The reverse process is performed when the Authentication Server sends the reply to the Supplicant.

At the end of the authentication process, the Authentication Server informs the Authenticator about the authorization status of the Supplicant, and in case of a successful authentication, it also sends the master key it has established with the Supplicant.

The technique used by the IEEE 802.1X protocol to control the communication during the authentication process employs the port based network access control mechanism. In particular, two logical access points to the network are defined, both controlled by the Authenticator: the uncontrolled port through which the authentication traffic (i.e. the EAP messages) is forwarded to the Authentication Server, and the controlled port that is disabled until the successful completion of the authentication procedure.

4.3. The IEEE 802.11i security protocol

The IEEE 802.11i protocol is the sixth amendment to the 802.11 standard designed to overcome its security weaknesses. It introduces the Robust Security Network Association (RSNA) that represents a logical connection between two 802.11 entities. This association is established at the end of a successful message exchange, named 4-Way Handshake, in which both parties prove their identity to each other by showing the ownership of a Pairwise Master Key (PMK). This key can be obtained as a Pre Shared Key (PSK) or after a successful port based authentication process, defined by the IEEE 802.1X protocol. In the latter case, the IEEE 802.11i standard defines how to use the authentication and authorization procedures in the context of IEEE 802.11 networks, assigning the Authenticator role to the Access Point (AP) and the Supplicant role to the wireless client (or station STA). Since in the IEEE 802.1X protocol the Supplicant authenticates itself to the network, the IEEE 802.11i standard requires a mutual EAP authentication method so that the network credentials can be verified by the Supplicant. The uncontrolled port of the Authenticator (AP) is used to forward authentication traffic between the Supplicant (STA) and the Authentication Server (AS); hence, the Authenticator executes only a passive role during the mutual authentication process performed by the Supplicant and the Authentication Server.

5. MobiSEC: a wireless mesh network security architecture

In this section we describe MobiSEC, the architecture by which we propose 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-step 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 cipher technique adopted. In the numerical evaluation, however, we used WEP [14] for two reasons: on the one hand it is the only cipher technique available for commercial wireless cards in ad hoc mode, which allowed us to implement MobiSEC in the MobiMESH testbed; on the other hand the utilization of WEP permits the robustness of the proposed solution to be proved, even in the presence of a weak cryptographic system.

We are currently implementing the CCMP algorithm for the IBSS operating mode [14], which is used by several mesh implementations to establish the backbone links and form a multi-hop wireless architecture.

5.1. 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 every wireless device currently available on the market implements the security functionalities described in the IEEE 802.11i protocol [14], 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 a 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.

Fig. 3 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 N_1 and N_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 N_1 and N_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.

5.2. 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-step approach is adopted, in which new nodes dynamically join the network as wireless clients and subsequently can upgrade their role, becoming wireless mesh routers by further authenticating to a Key Server.

Two major problems arise: on the one hand it is necessary to authenticate new mesh routers that join the network and provide them with 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 minimal impact on device mobility. To this end, we designed and implemented a key distribution 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 proof that the new node has the required credentials to become a mesh router.

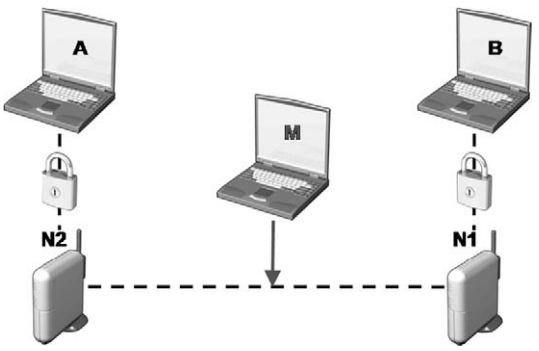

Fig. 3. Alice and Bob exchange data through the wireless mesh network. Mallory will be able to eavesdrop their data, unless a security system is implemented to protect the backbone link.

Fig. 4 shows the phases of the connection process performed by a new mesh router (namely, node N_2). Note that we illustrate only the most important messages exchanged between the network entities during the authentication process, while the whole procedure is detailed in the following.

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 protocol to complete a mutual authentication with the network and establish a security association with the entity to which it is physically connected through the execution of the 4-Way Handshake protocol (phase 1). In other words, during this phase N_2 performs all the activities as a generic wireless client to establish a secure channel with a mesh router (node N_1 in our example) that can forward its traffic securely over the wireless backbone. At the end of such phase, N_2 obtains the network parameters performing a DHCP request. In phase 2, N_2 establishes a secure connection with the Key Server (KS), using the TLS protocol [31] to obtain the necessary information that will be exploited 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.

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, at

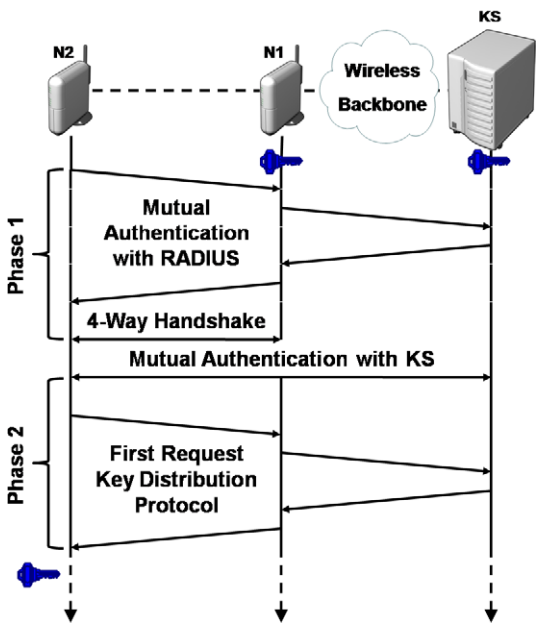

Fig. 4. Phases of the connection process performed by a new mesh router (node N_2). The depicted keys are used to encrypt backbone traffic.

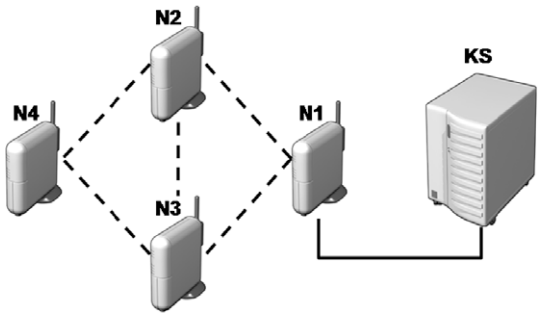

Fig. 5. Example of the proposed automated WMN configuration process. MobiSEC permits an automated and incremental configuration of the wireless mesh network.

the end of the successful authentication, an end-to-end secure channel is established between the Key Server and the mesh routers; the cryptographic material is then exchanged through such channel in a secure way.

To minimize the risks of using the same key for a long time, we propose two key distribution and regeneration protocols, described in Section 6, to create a new key when a pre-determined timeout expires. Both protocols require the synchronization of all mesh routers with a central server.

Fig. 5 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 N_1 can connect to the mesh network, since it is the only node that can complete the authentication with the Key Server and obtain from it the cryptographic material needed to set up an ad hoc and protected wireless link. The neighbors of N_1 (N_2 and N_3) 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 N_1 to produce the currently used cryptographic key. After having derived such key, both N_2 and N_3 will be able to reach each other, as well as node N_1, in ad hoc mode. Moreover they will be able to turn on their access interface through which they will provide to node N_4 a network connection towards the server.

6. Key distribution 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.

In both protocols, time is divided into *sessions*, whose duration is equal to the product of the number of keys used in a specific session and the key validity time, which is constant for every key of the session.

For the sake of clarity we illustrate the message exchanges and the performed operations considering a single-radio WMN, where all mesh routers are endowed with a single radio interface and communicate with each other using the same wireless channel. Multi-radio extensions are discussed in Section 6.4.

6.1. 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. In this protocol, each node maintains a list of n keys, which we refer to as the key list. Since commercial wireless boards commonly provide only 4 hardware registers to store cryptographic keys, in Section 7 we consider only key lists containing $n = 4$ keys. However, we underline that the proposed security architecture is general, and it is designed to manage key lists of arbitrary dimensions.

Fig. 6a shows in detail the message exchanges that occur between the mesh router and the Key Server. The function $E_k(\bullet)$ represents the symmetric cryptographic algorithm established between the two peers after a successful mutual authentication, and it is used to protect the secrecy and the integrity of the successive message exchanges. id_{req} and id_{node} represent respectively the request and the node identifier (i.e. the MAC address of the wireless card on which the request is sent). To improve the robustness of the protocol against reply attacks, all messages can contain further parameters (i.e. a timestamp and a nonce), but for the sake of brevity we did not include them in the figure.

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 routers that form the wireless backbone and the time when it was generated, the 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, n, and the maximum validity time of a generic key (the *timeout* parameter in Fig. 6a). Moreover, the key list validity starts when it is generated, i.e. at TS_{KL}. The node, based on the instant at which it joins the backbone (t_{now} in Fig. 6a), can identify the key among those in the list currently used by its peers, and its validity time (key_{id} and T_1), according to the following expression:

$$key_{id} = \left\lfloor \frac{t_{now} - TS_{KL}}{timeout} \right\rfloor + 1,$$
$$T_1 = key_{id} \cdot timeout - (t_{now} - TS_{KL}). \quad (1)$$

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 about 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

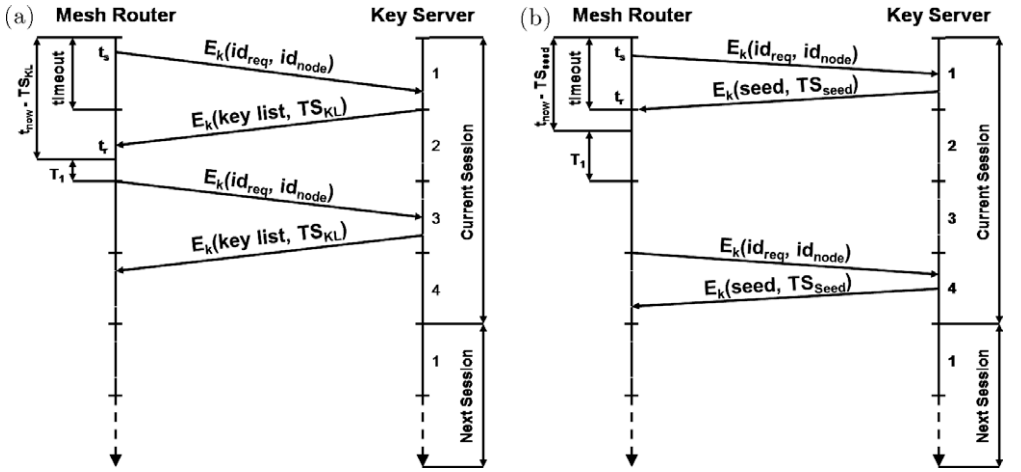

Fig. 6. Key distribution protocols: example message exchanges between the mesh router and the Key Server in the (a) Server Driven and (b) Client Driven protocols. $E_k(\bullet)$ represents the symmetric cryptographic function used to protect the security of the messages, whereas id_{req} and id_{node} represent the identifier of the request and of the node, respectively.

the key list cardinality and a correction factor, which can be estimated based on parameters such as the network load, the distance to the server, and the previous time to obtain the response.

In our architecture, such correction factor (c) is computed based on the time necessary to receive the response from the Key Server (Δt), which is estimated according to Eq. (2), where 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. So if a node takes a time (Δt in Eq. (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 obtain the response).

$$\Delta t = t_r - t_s,$$
$$\begin{cases} c = \left\lceil \frac{\Delta t - timeout}{timeout} \right\rceil & \text{if } \Delta t \geqslant timeout, \\ c = 0 & \text{if } \Delta t < timeout. \end{cases} \quad (2)$$

To illustrate how the correction factor is evaluated, let us refer again to the example message exchange shown in Fig. 6a; 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.

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 connected as generic wireless client through the wireless access network, while successive requests will be sent directly over the wireless backbone.

6.2. 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. In our implementation of MobiSEC we use MD5 [32] as hash function, which provides keys with length equal to 128 bit. Note that the proposed framework can easily be modified to use different hash functions and create keys with a different length.

Fig. 6b shows the message exchanges performed between the mesh router and the Key Server. As in the previous protocol, a generic mesh router, following 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 *timeout* parameter). Eq. (3) illustrates how to compute the number of times the mesh router must apply the hash function to synchronize its first key with that currently used by the other nodes (the r parameter), and its remaining validity time (T_1). The new key is computed as detailed in Eq. (4).

$$r = \left\lfloor \frac{t_{now} - TS_{seed}}{timeout} \right\rfloor + 1,$$
$$T_1 = r \cdot timeout - (t_{now} - TS_{seed}), \quad (3)$$

$$\begin{cases} key(r, seed) = hash(seed) & \text{if } r = 1, \\ key(r, seed) = hash(key(r-1, seed)) & \text{if } r > 1. \end{cases} \quad (4)$$

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, as proposed for example in [33];
- a maximum interval for the validity of the seed is set.

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 Fig. 6b the router performs such a 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 obtain the response before the session expiration.

The Server Driven and Client Driven protocols described above differ only for the cryptographic material provided by the Key Server and used by all mesh routers to generate session keys. The Server Driven protocol is the most secure of the two proposed protocols, since the session keys are generated randomly by the Key Server. Therefore, even if an adversary would be able to recover one of the session keys through a cryptanalytic attack, it would get access to the backbone network only for the remaining validity time of the broken key, since the successive key in the list would be uncorrelated with the previous one. However, the length of the messages exchanged by such protocol increases with increasing key list dimensions.

On the other hand, the Client Driven protocol imposes a lighter network overhead, since the length of the exchanged messages does not depend from the session duration, that is, from the number of keys generated using the same seed. Obviously, since all keys are generated through a hash-chain procedure, and are therefore correlated, an adversary that eventually recovered a key could gain access to the backbone network for the remaining validity time of the current session. However, the robustness of the Client Driven protocol against cryptanalytic attacks can be augmented as discussed above, i.e. through the concatenation of the seed and timestamps with a pre-shared secret known by each node.

6.3. Design and implementation of MobiSEC

Fig. 7 depicts the general architecture of the MobiSEC framework. We implemented the key distribution protocols as a client/server application using the OpenSSL library to authenticate and protect the connection that is established when a new node joins the wireless backbone network. In particular, each communication that takes place between a mesh router and the Key Server uses the TLS protocol both to authenticate the two entities and to protect the key material that is exchanged. The cryptographic material is communicated to the Key Switcher module that performs the tasks defined by our protocols to obtain and install the currently used key. We decided to implement this component as a kernel module to improve its responsiveness, especially under heavy network load conditions. In fact, the routing mechanism operating in kernel space can require a long execution time to manage the soft interrupts generated by the received packets, causing high level of delay in the scheduling of the user space processes. Therefore, implementing and running the module dedicated to deriving and installing the new key as a user space process may result in unpredictable scheduling delays, sometimes greater than the key validity time. On the other hand, such delay has a negligible effect on the client-side application (the Client Daemon module in Fig. 7), since the correction factor that is used to trigger the proactive request takes into account also this contribution.

6.4. Comments and security enhancements

In the following we discuss some design issues and security enhancements that can be used to improve MobiSEC.

6.4.1. Layer-2 encryption

In our implementation of MobiSEC we decided to use the encryption techniques provided by the MAC layer, since the most computationally complex operations are performed by the wireless card. Such solution has two

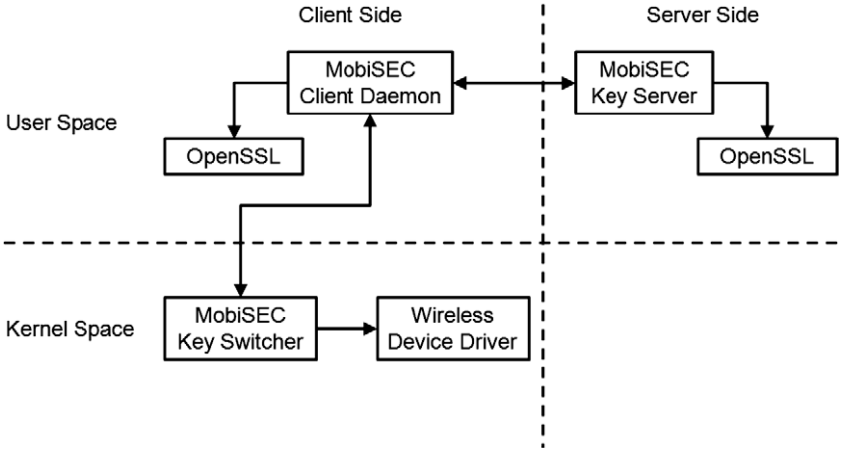

Fig. 7. MobiSEC architecture. The client-side application is installed on all mesh routers, whereas the server-side application is installed exclusively on the Key Server.

main advantages: on the one hand, the network performance is not impaired by executing such procedures; on the other hand a data-link layer encryption reduces the security requirements of the control and routing protocols.

6.4.2. Multi-radio extensions

The proposed architecture can easily be 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 messages format defined by the previous protocols so as to provide the additional information to the other end. In particular, the Key Server generates a different cryptographic information for each possible channel, whereas the mesh router requests and obtains the cryptographic information (key list or seed and type of the hash function) that is related only to the wireless channels on which its interfaces are set.

6.4.3. Synchronization issues

As we stated in the assumptions (Section 3), the synchronization of all mesh routers with the Key Server is a requirement for our architecture. However, in our tests we measured a synchronization difference among all nodes always smaller than a few milliseconds. Therefore, taking ample margins, we introduce a tolerance on the key validity of 2 s. This is obtained using cyclically three of the four hardware registers commonly provided by commercial wireless boards to install the cryptographic keys. The tolerance is realized setting the successive key of the sequence 2 s before the expiration of the current one and maintaining the previous key a further 2 s after its expiration. Such setting permits the obtaining of a performance that is very close to that achieved with a static key, as we will show in the next section, since both early and late nodes can properly decrypt the received frames.

6.4.4. Network partitioning

It may happen that the network is temporarily partitioned in two or more subnetworks due to interference or nodes malfunctioning, so that some mesh router can no longer connect to the Key Server. In this case, our architecture allows nodes inside each subnetwork to continue communicating among themselves using the current key. Furthermore, they periodically try to contact the Key Server to recover normal operation.

6.4.5. Detection techniques and certificate revocation

Finally, note that the authentication method based on certificate exchanges, used in our architecture, protects against *man in the middle* attacks, since all the certificates are signed by a trusted certification authority (CA), whose certificate is known by all network devices. Even if an adversary compromises a mesh router in order to obtain its certificate and gain access to the backbone network, it cannot impersonate or masquerade other network entities (such as mesh clients, the authentication server or the Key Server) as shown for example in [34,35], since the CA that releases the credentials is directly controlled by the WMN operator. Hence, in our architecture an adversary can steal the network identities only by breaking their private keys, which is computationally infeasible.

Moreover, MobiSEC can be coupled with detection techniques like those proposed in [8,23,36] to detect malicious nodes that could eventually enter the network. As soon as compromised wireless mesh routers are detected, these techniques can provide the identity of such nodes to the certification authority, so that this latter can transmit the certificate revocation list to the Key Server.

7. Performance evaluation

In this section we present the numerical results obtained in testing the proposed security framework with both the MobiMESH testbed and Network Simulator, considering different network scenarios. We compare the Server and Client Driven protocols with both a static key and an end-to-end approach. The first approach consists in securing the WMN with a fixed key; such scheme provides a bound to the performance that can be obtained by the proposed schemes, in terms of achievable throughput, delay and packet losses, while it is, obviously, a weak solution from the security point of view. The end-to-end approach consists in establishing a secure IPSec tunnel between the mesh client and the server; to this end we use the Openswan [37] implementation of IPSec for Linux.

We analyze the proposed protocols, varying the key validity time (the *timeout* parameter described in the previous section). Increasing the *timeout* value reduces the overhead introduced by the key distribution protocols, but at the same time this may reduce network security. In the following we show only the tests with $n = 4$ keys for each session, since we verified that the performance was not influenced significantly by this parameter. The *session* duration (for both the Server and Client Driven protocols) is therefore equal to $4 \cdot timeout$.

For each scenario we performed 10 independent measurements, achieving very narrow 0.95 confidence intervals, which we do not show for the sake of clarity. The total time on which we evaluated the performance both of real and simulated tests was equal to 600 s.

To prove the robustness of MobiSEC, as discussed before, 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) [38] with the KoreK improvements [39,40].

7.1. Experimental study

7.1.1. Full-mesh topology

We first considered the full-mesh network topology illustrated in Fig. 8, where each router is directly connected with the other two nodes (all nodes belong to the same ad hoc wireless cell). Router N_3 also acts as Key Server, so that both N_1 and N_2 send the key material request to N_3.

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 such results with those achieved on a radio link protected with a static key and by establishing an encrypted IPSec tunnel between each

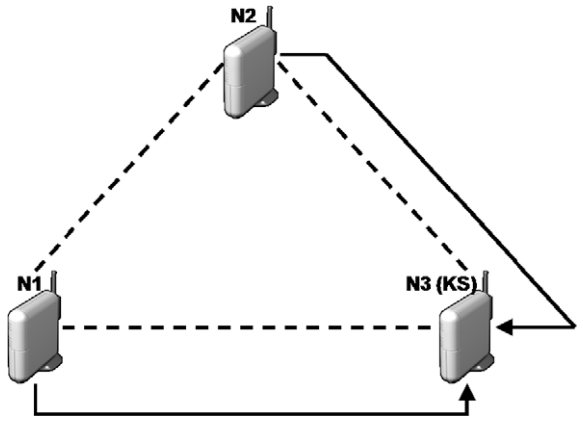

Fig. 8. Full-mesh topology. A data transfer is performed between nodes N_1 and N_3. Although N_3 also acts as Key Server, the connection among the three nodes remained available in all the tests we performed.

pair of nodes. TCP traffic was generated between mesh routers N_1 and N_3 with the D-ITG traffic generator [41].

Fig. 9 shows the throughput achieved by both the proposed protocols, as a function of the key validity time. The maximum throughput is very similar for both the Server and Client Driven protocols, since the computation of the key sequence performed by the Client Driven protocol does not impair the achievable throughput. Furthermore, note that such throughput is very close to the bound provided by the static key technique. On the other hand, the IPSec solution achieves a lower performance, which is mainly due to the fact that layer-2 encryption, used by all the other considered protocols, is directly supported in the wireless card hardware.

At the same time, we tested the availability of the Key Server, and we verified that all mesh routers could remain connected even in the presence of a high network load.

In the same scenario we measured the effectiveness of the key tolerance mechanism described in Section 6, disabling it and measuring the protocols' performance. The corresponding numerical results are shown in Fig. 10, and the performance improvement introduced by implementing the tolerance mechanism is evident (see Fig. 9 for a comparison).

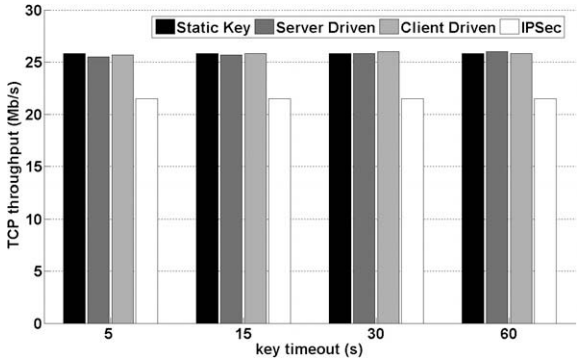

Fig. 9. TCP throughput measured in the full-mesh network scenario for different key distribution protocols and key validities, using the tolerance on the key validity time.

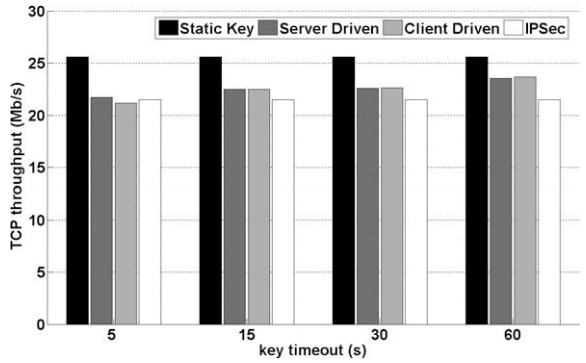

Fig. 10. TCP throughput measured in the full-mesh network scenario for different key distribution protocols and key validities, disabling the tolerance on the key validity time.

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 the wireless link between nodes N_2 and N_3. The transmission rate was set to 10 Mb/s and several data transfer sessions were performed, each with a duration ranging in the 2–12 min interval. We observe that the choice of the number of keys used in a session, n, has no impact on the packet loss, since we measured a negligible value of such performance figure in all our experiments.

7.1.2. Strength analysis

Strength analysis has been carried out in the same network scenario to evaluate how much our solution increases the overall security, even when used with a weak cryptographic mechanism like WEP. For both the proposed protocols we set the key timeout and the session duration to 60 and 240 s, respectively. Such analysis was performed sniffing the traffic transmitted between N_1 and N_3 and then applying a crypto-analytic attack with the aircrack-ng tool.

Table 1 reports the outcome of such 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 derive the key was significantly larger than the theoretical number indicated in [39,40]. 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 s considering an 802.11 packet and the theoretical throughput of an 802.11a/g wireless link. Therefore, setting the maximum key validity time to 60 s turns out to be a relatively 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) had no effect on the results of the attacks performed against our protocols: in both cases aircrack-ng failed to recover the keys used to encrypt the frames. The slightly longer execution time taken by the tool to crack the static key when the packet-gathering time is equal to 600 s 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 and 240 s, respectively. The packet-gathering time varied from 60 to 600 s.

Protocol	Packet-gathering time (s)		
	60	240	600
Fudge factor = 2			
Static key	Failed	Failed	Cracked (5 s)
Server Driven	Failed	Failed	Failed
Client Driven	Failed	Failed	Failed
Fudge factor = 4			
Static key	Failed	Failed	Cracked (7 s)
Server Driven	Failed	Failed	Failed
Client Driven	Failed	Failed	Failed

7.1.3. Multi-hop topology

We then considered the multi-hop network scenario illustrated in Fig. 11, where solid and dashed lines represent wired and wireless links, respectively. All nodes were equipped with two wireless interfaces, which were set on orthogonal channels so that each mesh router was connected only to the previous and the subsequent node. All mesh routers run the client-side application of the Client and Server Driven protocol, whereas node KS acted only as Key Server.

We performed a data transfer between nodes N_1 and KS using the D-ITG traffic generator, and we measured the maximum TCP throughput of the two proposed protocols; then, we compared such results with those obtained using a static key solution and establishing an encrypted IPSec tunnel between the nodes N_1 and KS. The results, reported in Fig. 12, confirm the trend of the previous network scenario: the tolerance introduced on the key validity time permits an improvement in the strength of the proposed scheme without reducing consistently the overall throughput.

In the same scenario we evaluated the Round Trip Time (RTT), setting the packet size to 1500 bytes. Table 2 shows the results (expressed in milliseconds) that we measured setting the key timeout and the session duration to 30 and 120 s, respectively. The low value of the RTT's standard deviation suggests that our solution guarantees a correct operation even for real-time multimedia applications without introducing perceptible alterations in the transmitted stream.

Since all the results we measured show that the IPSec solution performs consistently worse than the proposed protocols, for the sake of brevity in the following we do not report the results obtained with such technique.

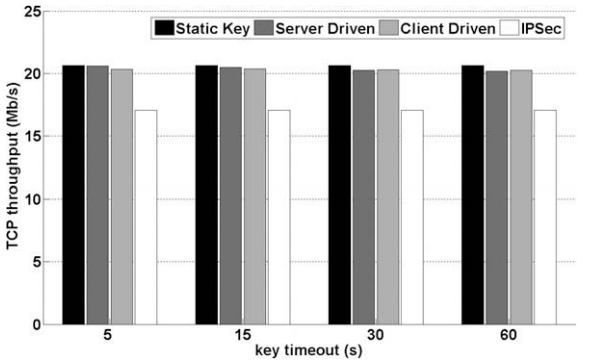

Fig. 12. TCP throughput measured in the multi-hop network scenario for different key distribution protocols and key validities, using the tolerance on the key validity time.

Table 2
Multi-hop topology: Round Trip Time measured in ms for a TCP connection established between nodes N_1 and KS.

Parameter	Static key	Client	Server
Average RTT	8.2	8.5	8.4
Minimum RTT	7.9	8.3	8.2
Maximum RTT	8.4	8.7	8.6
RTT standard deviation	1.2	1.2	1.2

7.1.4. Broadband office networking

Finally, to obtain a sense of the quality of the proposed approaches in a real life scenario, we measured the performance of MobiSEC in a Wireless Mesh Network that covered the offices of the Telecommunications Network Group in our Department. To this end, we have constructed a WMN with 10 mesh routers, placed as shown in Fig. 13a; Fig. 13b shows the logical topology obtained with such node placement, as well as the location of the Key Server and the mesh client used in this experiment.

On all mesh routers we installed the client-side of the MobiSEC application and the UniK implementation of the OLSR routing protocol [42]. Node KS was configured to host all network services, namely user and node authentication, key distribution and time synchronization. A DHCP server that provides the necessary network parameters was also installed on the same node.

The arrow depicted in Fig. 13b represents the data transfer we performed to measure the TCP throughput on the path N_1–N_2–N_3–N_4–N_7–N_8–N_9. On the paths N_3–N_4–N_5 and N_6–N_7–N_8 we also set a background data

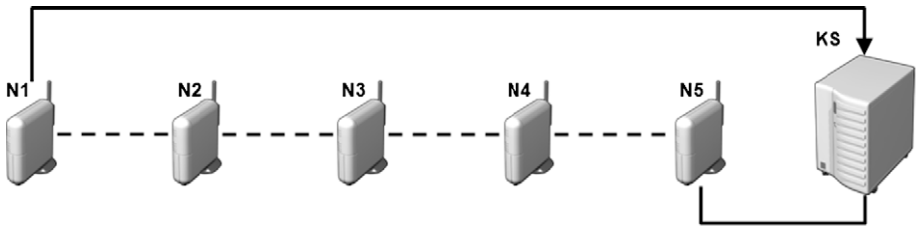

Fig. 11. Multi-hop topology. A multi-hop data transfer between nodes N_1 and KS is performed to measure the network performance.

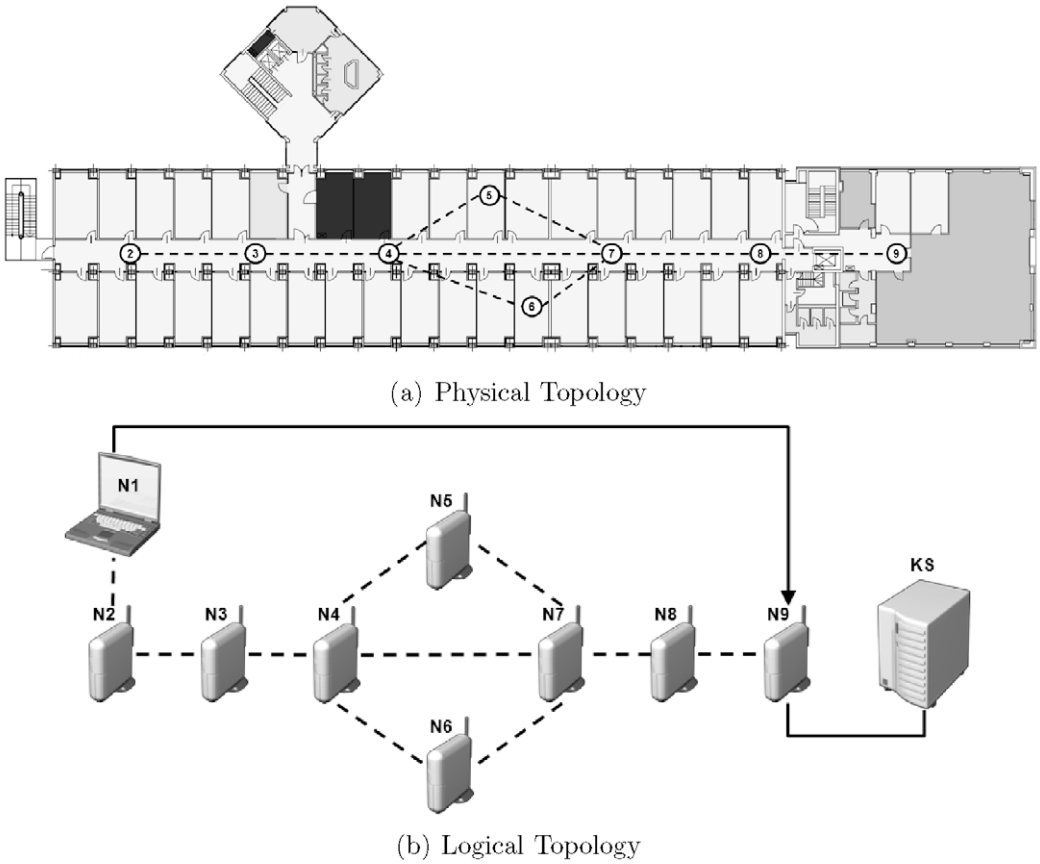

Fig. 13. Broadband office networking. The arrow represents the measured TCP data transfer. On the paths N_3—N_4—N_5 and N_6—N_7—N_8 a UDP data transfer performed at the constant rate of 2 Mbit/s.

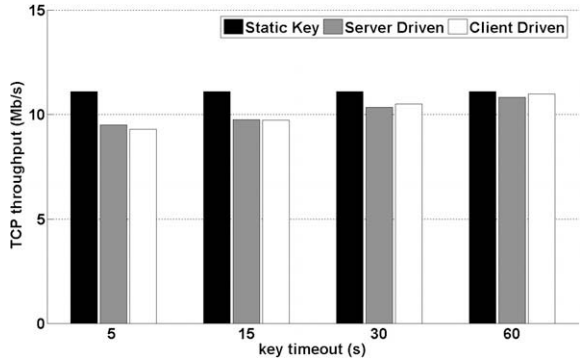

Fig. 14. TCP throughput measured in the Broadband Office Networking scenario for different key distribution protocols and key timeout.

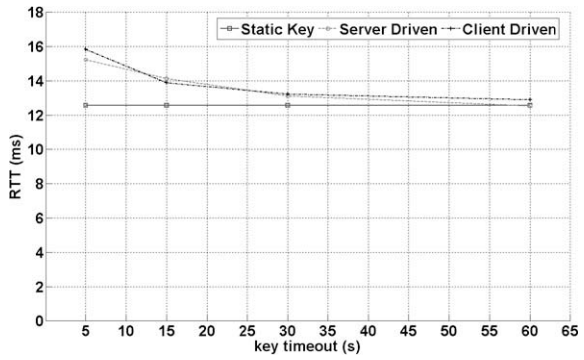

Fig. 15. Average Round Trip Time as a function of the key timeout measured in the topology of Fig. 13.

transfer at the constant rate of 2 Mbit/s to simulate a real network utilization.

Numerical results are shown in Fig. 14, and confirm the trend observed in the previous network scenarios. The absolute performance is lower than in the multi-hop topology due to the increased traffic of background communications and routing messages, which are forwarded by each mesh router.

In the same scenario we also evaluated the mean Round Trip Time. For each security scheme we varied the key validity time, and the numerical results are shown in Fig. 15. The similarity with the values obtained with a static key confirms that our security solution does not introduce significant degradations in such performance figures.

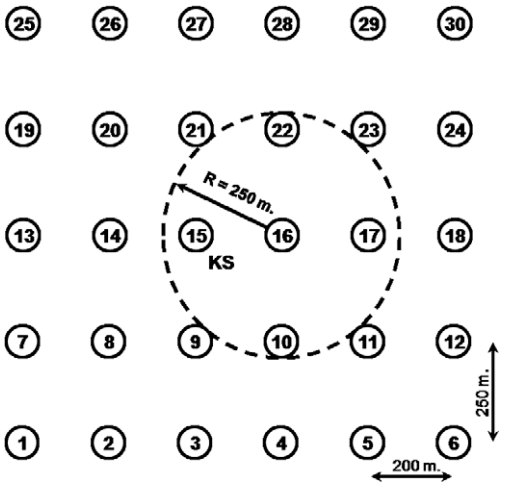

Fig. 16. Grid topology. A TCP connection is established between nodes 1 and 30. The Key Server (KS) is located at node 15.

7.2. Simulation study

To evaluate the proposed architecture in large-scale network scenarios, we further implemented the MobiSEC architecture extending the Network Simulator.

We considered two different network topologies: the same multi-hop topology illustrated in Fig. 11, and the grid topology illustrated in Fig. 16, where 30 nodes are placed over a 1000 m $\times$ 1000 m square area; in this latter topology, all nodes were spaced 200 m horizontally and 250 m vertically.

The maximum channel capacity was set to 54 Mbit/s. All nodes use the same wireless channel since ns v.2 does not support natively multi-channel or multi-interface wireless nodes. Moreover, we used the UM-OLSR implementation of the OLSR routing protocol [43].

We measured the average throughput of a TCP connection between two nodes, varying the key validity time and the session duration. In particular, for the multi-hop topology the TCP connection was established between nodes N_1 and KS, whereas in the grid network it was established between the bottom left and top right nodes (nodes 1 and 30 of Fig. 16), while node 15 acted as Key Server. TCP NewReno was used for TCP sources, and receivers implemented the Delayed ACKs algorithm. The Maximum Segment Size was equal to 1500 bytes.

Fig. 17 shows the TCP throughput obtained in these two topologies as a function of the key timeout. As expected, the performance decreases when the key timeout is reduced. This trend is more evident in the grid topology, since the greater number of messages exchanged by the mesh routers with the Key Server causes a larger number of collisions. However, for a key timeout of 60 s, the Server and Client Driven protocols perform close to the bound provided by the static key approach. Since we have shown in the previous scenarios that such setting is relatively conservative from a security point of view, we can expect that MobiSEC performs close to the optimum also in this scenario with several nodes.

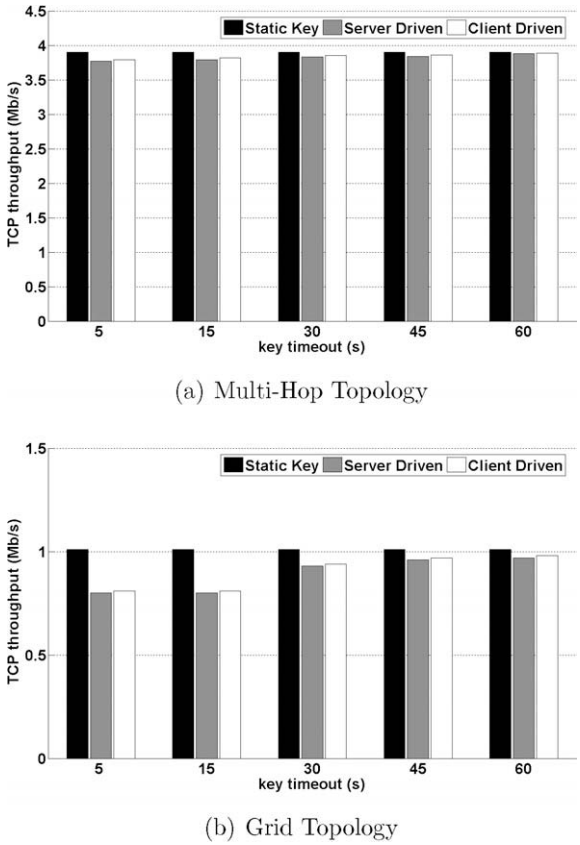

Fig. 17. TCP throughput measured using ns v.2 in (a) the multi-hop topology and in (b) the grid scenario.

Finally, we observe that the discrepancy between the TCP throughput values shown in Figs. 12 and 17a is mainly due to the different configurations that exist between real and simulated scenarios. In these latter, as discussed above, we were forced to use a unique wireless channel, since Network Simulator does not provide a support for multi-channel or multi-interface wireless nodes. However, even though the absolute values of the simulated results cannot be compared to the testbed measurements, they exhibit the same trend, thus confirming the validity of the measurement campaign we conducted.

8. Conclusion

In this paper we proposed MobiSEC, a novel security architecture tailored for wireless mesh networks. MobiSEC addresses 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, comparing its performance with that of existing schemes, viz.: static key encryption and end-to-end IPSec tunnel solutions. Furthermore, we simulated the behavior of MobiSEC in large-scale network instances using Network Simulator.

Numerical results show that MobiSEC offers secure network services to both mesh users and routers with negligible impact on network performance, since it achieves high transmission rates and low latencies, therefore representing an effective solution for wireless mesh networking.

Acknowledgment

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

References

[1] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.
[2] R. Bruno, M. Conti, E. Gregori, Mesh networks: commodity multihop ad hoc networks, IEEE Communications Magazine 43 (3) (2005) 123–131.
[3] N. Ben Salem, J.-P. Hubaux, Securing wireless mesh networks, IEEE Wireless Communications 13 (2) (2006) 50–55.
[4] C. Adjih, D. Raffo, P. Mühlethaler, Attacks against OLSR: distributed key management for security, in: Proceedings of the First OLSR Interop and Workshop, August 2005.
[5] W. Stallings, Cryptography and Network Security, fourth ed., McGraw-Hill, 2003.
[6] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Mühlethaler, D. Raffo, Securing the OLSR protocol, in: Proceedings of the IFIP Med-Hoc-Net, 2003.
[7] D. Raffo, C. Adjih, T. Clausen, P. Mühlethaler, An advanced signature system for OLSR, in: SASN'04: Proceedings of the Second ACM Workshop on Security of Ad Hoc and Sensor Networks, 2004, pp. 10–16.
[8] N. Komninos, D. Vergados, C. Douligeris, Detecting unauthorized and compromised nodes in mobile ad hoc networks, Ad Hoc Networks 5 (3) (2007) 289–298.
[9] H. Luo, P. Zerfos, J. Kong, S. Lu, L. Zhang, Self-securing ad hoc wireless networks, in: ISCC 2002: Proceedings of the Seventh International Symposium on Computers and Communications, July 2002, pp. 567–574.
[10] N. Milanovic, M. Malek, A. Davidson, V. Milutinovic, Routing and security in mobile ad hoc networks, IEEE Computer 37 (2) (2004) 61–65.
[11] L. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Network 13 (6) (1999) 24–30.
[12] O. Cheikhrouhou, M. Laurent-Maknavicius, H. Chaouchi, Security architecture in a multi-hop mesh network, in: SAR 2006: Proceedings of the Fifth Conference on Safety and Architectures Networks, June 2006.
[13] R. Fantacci, L. Maccari, T. Pecorella, F. Frosali, A secure and performant token-based authentication for infrastructure and mesh 802.1X networks, in: Infocom'06 Poster Session, April 2006.
[14] IEEE Standard 802.11i, Medium Access Control (MAC) Security Enhancements, Amendment 6, IEEE Computer Society, 2004.
[15] A. Capone, S. Napoli, A. Pollastro, MobiMESH: an experimental platform for wireless mesh networks with mobility supports, in: WiMESHNets'06: Proceedings of the First ACM Workshop on Wireless Mesh: Moving Towards Applications, ACM, August 2006.
[16] Vint Project U.C. Berkeley/LBNL ñs-2 Network Simulator (ver. 2), <http://www.isi.edu/nsnam/ns/>.
[17] IEEE Standard 802.1X, Port-based Network Access Control, IEEE Computer Society, 2004.
[18] A. Mishra, W.A. Arbaugh, An initial security analysis of the IEEE 802.1X standard, UM Computer Science Department, Technical Report CS-TR-4328, February 2002.
[19] M. Kassab, A. Belghith, J.-M. Bonnin, S. Sassi, Fast pre-authentication based on proactive key distribution for 802.11 infrastructure networks, in: WMuNeP'05: Proceedings of the First ACM Workshop on Wireless Multimedia Networking and Performance Modeling, ACM, 2005, pp. 46–53.
[20] A.R. Prasad, H. Wang, Roaming key based fast handover in WLANs, in: WCNC'05: Proceedings of the IEEE Wireless Communications and Networking Conference, vol. 3, March 2005, pp. 1570–1576.
[21] Y. Zhang, Y. Fang, Arsa: an attack-resilient security architecture for multihop wireless mesh networks, IEEE Journal on Selected Areas in Communications 24 (10) (2006) 1916–1928.
[22] Y. Fu, J. He, R. Wang, G. Li, Mutual authentication in wireless mesh networks, in: ICC'08: Proceedings of the International Conference on Communications, May 2008, pp. 690–1694.
[23] B. Parno, A. Perrig, V. Gligor, Distributed detection of node replication attacks in sensor networks, in: Proceedings of the Symposium on Security and Privacy, May 2005.
[24] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, SPINS: security protocols for sensor networks, Wireless Networks 8 (5) (2002) 521–534.
[25] G. Xu, L. Iftode, Locality driven key management architecture for mobile ad-hoc networks, IEEE International Conference on Mobile Ad-hoc and Sensor Systems (2004) 436–446.
[26] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, E.M. Belding-Royer, A secure routing protocol for ad hoc networks, in: Proceedings of the 10th IEEE International Conference on Network Protocols, November 2002, pp. 78–87.
[27] Y.-C. Hu, A. Perrig, D.B. Johnson, Ariadne: a secure on-demand routing protocol for ad hoc networks, Wireless Networks 11 (1–2) (2005) 21–38.
[28] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Extensible Authentication Protocol (EAP), RFC 3748, June 2005.
[29] D. Stanley, J. Walker, B. Aboba, Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs, RFC 4017, March 2005.
[30] C. Rigney, S. Willens, A. Rubens, W. Simpson, Remote Authentication Dial in User Service (Radius), RFC 2865, June 2000.
[31] T. Dierks, C. Allen, The TLS Protocol Version 1.0, RFC 2246, January 1999.
[32] R. Rivest, The md5 Message-digest Algorithm, RFC 1321, April 1992.
[33] F. Stajano, R. Anderson, The resurrecting duckling: security issues for ad-hoc wireless networks, in: Proceedings of the Seventh International Workshop on Security Protocols, Springer-Verlag, 2000, pp. 172–194.
[34] M. Kumar, New remote user authentication scheme using smart cards, IEEE Transactions on Consumer Electronics 50 (2) (2004) 597–600.
[35] J.J. Shen, C.W. Lin, M.S. Hwang, A modified remote user authentication scheme using smart cards, IEEE Transactions on Consumer Electronics 49 (2) (2003) 414–416.
[36] S. Hakami, Z. Zaidi, B. Landfeldt, T. Moors, Detection and identification of anomalies in wireless mesh networks using principal component analysis (pca), in: The International Symposium on Parallel Architectures, Algorithms, and Networks, 2008, pp. 266–271.
[37] Openswan Project, <http://www.openswan.org/>.
[38] S. Fluhrer, I. Mantin, A. Shamir, Weaknesses in the key scheduling algorithm of RC4, Lecture Notes in Computer Science 2259 (January) (2001) 1–24.
[39] W.A. Arbaugh, N. Shankar, Y.C.J. Wan, K. Zhang, Your 802.11 wireless network has no clothes, IEEE Wireless Communications 9 (6) (2002) 44–51.
[40] A. Stubblefield, J. Ioannidis, A.D. Rubin, A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP), ACM Transactions on Information and System Security (TISSEC) 7 (2) (2004) 319–332.
[41] A. Botta, A. Dainotti, A. Pescapé, Multi-protocol and multi-platform traffic generation and measurement, in: Infocom'07 DEMO Session, vol. 45, May 2007, pp. 526–532.
[42] Olsrd Project, <http://www.olsr.org/>.
[43] Francisco J. Ros, Um-OLSR Project, 2005, <http://masimum.dif.um.es/um-olsr/html/>.

Fabio Martignon received the Laurea and the Ph.D. degree in telecommunication engineering from the Politecnico di Milano in October 2001 and May 2005, respectively. He is now an assistant professor in the Department of Information Technology and Mathematical Methods at the University of Bergamo. His current research activities include routing and MAC for multi-hop wireless networks, network planning, congestion control and QoS routing over IP networks.

Stefano Paris received the M.S. degree in Computer Engineering from the University of Bergamo in May 2007.
He is now a Ph.D. student at the Dipartimento di Elettronica e Informazione of the Politecnico di Milano.
His research interests include topics related to security architectures, reputation frameworks and detection schemes for wireless mesh and community networks.

Antonio Capone is an Associate Professor at the Information and Communication Technology Department (Dipartimento di Elettronica e Informazione) of the Technical University of Milan (Politecnico di Milano). His expertise is on networking and main research activities include protocol design (MAC and routing) and performance evaluation of wireless access and multi-hop networks, traffic management and quality of service issues in IP networks, network planning and optimization. On these topics he has published more than one hundred peer-reviewed papers in international journals and conference proceedings, and holds several patents. He received the M.S. and Ph.D. degrees in electrical engineering from the Politecnico di Milano in 1994 and 1998, respectively. In 2000 he was a visiting professor at UCLA, Computer Science department. He currently serves as editor of the Wiley Journal of Wireless Communications and Mobile Computing and the Elsevier Journal of Computer Networks. He served as guest editor of the Special Issue of the IEEE Wireless Communications magazine on 3G/4G/WLAN/WMAN Planning and Optimization, the Special Issue of the Elsevier Ad Hoc Networks journal on Recent research directions in wireless ad hoc networking and as member of the technical program committee of several international conferences. He is currently involved in the scientific and technical activities of several national and European research projects, and he leads several industrial projects. He is a Senior Member of the IEEE (Communications, Computer and Vehicular Technology societies).