SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks **00**: 1–17 (0000)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.0000

DSA-Mesh: a Distributed Security Architecture for Wireless Mesh Networks

Fabio Martignon[1], Stefano Paris[2] and Antonio Capone[2]

[1]*Department of Information Technology and Mathematical Methods, University of Bergamo*
E-mail: fabio.martignon@unibg.it
[2]*Department of Electronics and Information, Politecnico di Milano*
E-mail: {paris, capone}@elet.polimi.it

Summary

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 DSA-Mesh, a fully distributed security architecture that provides access control for mesh 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 backbone of the WMN.
DSA-Mesh exploits the routing capabilities of mesh routers: after connecting to the access network as generic wireless clients, new mesh routers authenticate to a key management service (consisting of several servers) implemented using threshold cryptography, and obtain a temporary key that is used both to prove their credentials to neighbor nodes and to encrypt all the traffic transmitted on wireless backbone links.
A key feature in the design of DSA-Mesh is its independence from the underlying wireless technology used by network nodes to form the backbone. Furthermore, DSA-Mesh allows seamless mobility of mesh routers. Since it is completely distributed, DSA-Mesh permits to deploy automatically and incrementally large wireless mesh networks, while increasing, at the same time, the robustness of the system by eliminating the single point of failure typical of centralized architectures.
DSA-Mesh has been implemented in Network Simulator, and extensive simulations have been performed in large-scale network scenarios, comparing it to a static key approach and to a centralized architecture where a single key server is deployed. Numerical results show that our proposed architecture considerably increases the WMN security and reliability, with a negligible impact on the network performance, thus representing an effective solution for wireless mesh networking.

KEY WORDS: Wireless Mesh Networks, Authentication, Security, Threshold Cryptography, Session Secret Distribution

1. Introduction

The Wireless Mesh Network (WMN) paradigm has recently emerged as a viable and cost-effective means to deploy all-wireless network infrastructures [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, 7, 8, 9, 10, 11], they are not suitable for WMNs, which 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 user 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 focus on backbone area security by proposing DSA-Mesh, a novel and fully Distributed Security Architecture for Wireless Mesh Networks, which provides a security framework for the mesh backbone, that is, access control for mesh routers as well as security and integrity of all data communications that occur in the WMN; this is achieved with layer-2 encryption through the utilization of a shared key whose delivery is assured by a key distribution protocol.

DSA-Mesh exploits the routing capabilities of wireless mesh routers, adopting a two-step approach: (1) in the first step, new nodes perform the authentication process with the nearest mesh router, like generic wireless clients; (2) in the second step, these nodes can upgrade their role in the network (becoming mesh routers) by further authenticating to a key management service, which consists of several servers, obtaining a temporary key with which all traffic is encrypted. Such step is implemented using threshold cryptography, which permits the distribution of trust in the key management service, allowing n mesh nodes* to share the ability to perform a cryptographic operation (e.g., creating a digital signature), so that any t nodes can perform this operation jointly, whereas it is infeasible for at most $t-1$ nodes to do so, even by collusion.

*The terms *node* and *router* will be used interchangeably

Furthermore, in this paper we propose two distributed protocols: the first implements a proactive secret delivery to generic mesh routers, whereas the second, based on Shamir's no-key protocol, permits to distribute the session secret to all core nodes, so that each node in the core set agrees with all other core nodes on the same secret.

A key feature in the design of DSA-Mesh is its independence from the underlying wireless technology used by network nodes to form the backbone. Furthermore, DSA-Mesh permits seamless mobility of mesh routers, which can roam freely around the backbone network after getting the key material from the nodes that implement the *key service*, since all other mesh routers create the temporary key using the same information. Finally, we underline that DSA-Mesh is fully distributed and can therefore be used to deploy automatically and incrementally large wireless mesh networks. In this regard, DSA-Mesh can support the development of Wireless Community Networks [14], where groups of people use the Wireless Mesh technology to form a dynamic, self-organizing, and citizens-owned pervasive communication infrastructure.

We extended Network Simulator (ns v.2) [15] implementing the DSA-Mesh architecture, and we performed extensive simulations in several realistic and large-scale network scenarios, comparing DSA-Mesh both with a static approach (which consists in using a fixed key to protect the WMN), as well as with a centralized architecture, proposed in [16], where a single Key Server is deployed. The fixed key approach provides an upper bound in terms of achievable throughput, delay and packet losses, while the second approach is useful to gauge the performance gap between DSA-Mesh and a centralized architecture.

Numerical results show that DSA-Mesh considerably increases the wireless mesh network security, as well as the system's robustness against node failures, 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 DSA-Mesh, a fully distributed security architecture for the backbone area of a WMN;
- the proposition of two novel and efficient secret delivery and secret agreement protocols;
- a thorough evaluation of the proposed architecture in several realistic network scenarios.

DOI: 10.1002/sec

The paper is structured as follows: Section 2 discusses related work. Section 3 provides an overview of the cryptographic primitives and algorithms used in our architecture. Section 4 briefly presents the centralized security architecture we proposed in [16], which will be compared to DSA-Mesh in the Numerical Results Section. Section 5 describes the proposed distributed security framework and the key distribution protocols. Section 6 discusses numerical results that show the effectiveness of our solution in a set of realistic network scenarios. Finally, conclusions are presented in Section 7.

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 [17, 18, 19], 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, 20, 21].

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 [22] 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.

The authors of [23] 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 prevent and detect 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 to other network agents, and passes it to the neighboring nodes for further decisions. In [24], two protocols are defined to detect replicated nodes by distributing the information about each node's identity and geographical position 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.

Other works investigate the use of threshold cryptography to achieve high fault tolerance against network partitioning. In [9] and [25] two different approaches are presented to allow specific coalitions of devices to act together as a single certification authority, whereas in [26] a hierarchical key management architecture is proposed to obtain an efficient establishment of distributed trust. Capkun et al. [27] propose a fully self-organized public-key management scheme that, similarly to the PGP scheme, does not rely on any trusted authority to perform the authentication of other peer nodes: each network node is its own certification authority and issues certificates to other nodes; the authentication procedure is performed via trust chains of certificates. The public key management schemes proposed in [28] and [29] further enhance the security of the distributed approaches like those presented in the above works,

DOI: 10.1002/sec

by using proactive secret sharing and fast verifiable share redistribution techniques which permit to update periodically the secret shares.

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. On the other hand, the centralized architecture proposed in [16] (MobiSEC), provides both access control for mesh users and routers with a negligible impact on the network performance. We will revise in more detail the MobiSEC architecture in Section 4.

Finally, we underline that none of the above solutions 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 fully distributed framework that copes with the security problems of the backbone area of a WMN, maintaining a high level of compatibility with current wireless security standards without impacting, at the same time, on the WMN performance.

3. Cryptographic Primitives and Algorithms

In this Section we introduce the cryptographic primitives and algorithms used in our architecture to distribute the Key Server functionalities among a group of mesh routers.

We first introduce the Shamir Secret Sharing algorithm, which is used to share the *key service* private key among a set of core mesh routers; then, we provide an overview of the Threshold Signature Scheme, which is used by all generic mesh routers to prove the authenticity of the messages signed by the core mesh routers.

3.1. Shamir Secret Sharing Algorithm

In [30], Shamir proposes a method to share a secret among a group of parties. In an (n, t) threshold sharing scheme, a secret S is divided into n secret shares, but only t out of n pieces are necessary to recover the original secret.

The scheme is based on the following property: if $f(x) = S + \sum_{i=1}^{t-1} a_i x^i$ is a polynomial of order $t-1$ whose coefficients a_i are chosen over a finite field Z_q (where q is a large prime) and $a_0 = S$, then only t distinct points $\{(x_i, f(x_i))\}$ are necessary to recover the secret S, while $t-1$ or fewer points provide no information about the shared secret. The method used to recover the secret is known as *Lagrange interpolation*, which is briefly sketched in the following.

Let $C = \{s_1, s_2, ..., s_n\}$ be the set of the n secret shares, where $s_i = f(i) \, mod \, q$, and let A be any subset of C whose cardinality is equal to t ($A \subset C, |A| = t$). The secret S can then be recovered from A, according to the following equation:

$$\begin{aligned} l_i(x) &= \prod_{j \in A, j \neq i} \frac{x-j}{i-j} \\ k_i &= l_i(0) \cdot s_i = l_i(0) \cdot f(i) \, mod \, q \\ S &= \sum_{i \in A} k_i \, mod \, q = \sum_{i \in A} l_i(0) \cdot s_i \, mod \, q \end{aligned} \tag{1}$$

3.2. Threshold Signature Scheme

Threshold signature schemes permit to verify the authenticity of the signature applied to a message by a coalition of t out of n parties without revealing the private key.

In an RSA signature scheme [31], the private exponent d of the *key service* private key ($K_k^{-1} = \langle d, N \rangle$) can be shared by n parties.

The signature of any message m, where $h(m)$ represents the digest of m (computed using a one-way hash function), can be recovered by collecting t out of n partial signatures and multiplying them according to the following expression:

$$\begin{aligned} &\prod_{i=1}^{t} h(m)^{k_i} \, mod \, N = \\ &= h(m)^{\sum_{i=1}^{t} k_i} \, mod \, N = \\ &= h(m)^{\sum_{i=1}^{t} l_i(0) \cdot q(i)} \, mod \, N = \\ &= h(m)^d \, mod \, N \end{aligned} \tag{2}$$

Finally, to verify the authenticity of the message, the node has to raise the previous signature to the public exponent e, and compare the obtained result with the hash value of the message, according to expression (3).

$$\begin{aligned} &(h(m)^d \, mod \, N)^e \, mod \, N = \\ &= h(m)^{de} \, mod \, N = \\ &= h(m) \end{aligned} \tag{3}$$

DOI: 10.1002/sec

4. Overview of the MobiSEC Architecture

Having provided an overview of the cryptographic primitives and algorithms used in DSA-Mesh, in this Section we briefly introduce MobiSEC, the centralized architecture we proposed in [16], which will be used as term of comparison for DSA-Mesh in the Numerical Results Section.

The two architectures adopt a similar approach to protect the backbone of a WMN, that is, all mesh routers obtain the same temporary secret which is used both to prove their credentials to neighbor nodes and to encrypt all the traffic transmitted on the wireless backbone links. However, DSA-Mesh is a completely distributed architecture, since it distributes the Key Server functionalities among a group of core nodes.

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

Figure 1 shows the three phases of the connection process performed by a new mesh router (namely, node N_2). When N_2 wants to connect to the mesh network, it scans all radio channels to detect a mesh router already connected to the wireless backbone, which is therefore able to provide access to all network services (including authentication and key distribution). Let N_1 be such router. After connecting to N_1, N_2 can perform the tasks described by the IEEE 802.11i 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). 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 [32], to obtain the necessary information that will be used to generate the current key used by all mesh routers to encrypt all the traffic transmitted on the mesh backbone. In particular, the device can connect to the wireless backbone in a secure way and begin executing the routing and access functions (phase 3).

During phase 2, mesh routers also perform a second authentication, based on the TLS protocol.

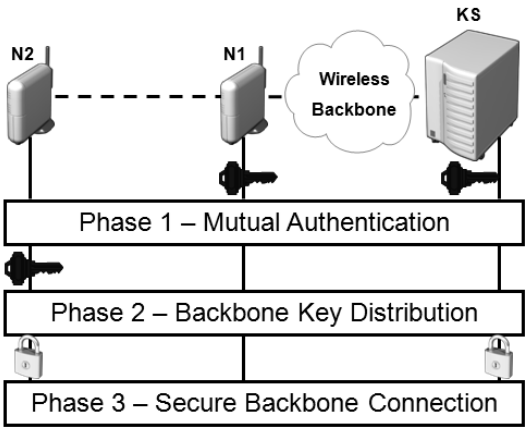


Fig. 1. Centralized security architecture: phases of the connection process performed by a new mesh router (node N_2). The depicted keys are used to encrypt backbone traffic.

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.

5. DSA-Mesh: a distributed security architecture for WMNs

Centralized solutions, like the one reviewed in the previous Section, can exhibit lower costs than distributed approaches; however, they are characterized by a single point of failure (the Key Server), which can be exploited by an adversary to attack and subvert the whole network.

A naive solution to improve the overall availability and robustness of the centralized architecture could be represented by having more nodes that implement the functions provided by the Key Server. This solution, however, is exposed to two serious problems: on the one hand, all Key Servers should agree on the secret provided to every mesh router in each session; on the other hand, the system security would be greatly reduced, since an adversary should compromise at most one server to produce fake seeds or key list responses.

To overcome both such problems we therefore propose DSA-Mesh, a novel and fully distributed security architecture that uses two distributed protocols based on threshold cryptography and Shamir's no-key protocol.

DOI: 10.1002/sec

5.1. Assumptions and Notation

In order to specify the WMN scenario we are dealing with, we adopt the following definitions and assumptions:

- All nodes authorized to join the wireless backbone have a public/private pair of keys and a certificate that binds the public key to the node identity. The certificate is signed by a trusted certification authority, whose certificate is known by all mesh routers.
- The set R of all mesh routers is divided into two subsets: $C \subset R$ is composed by the core nodes, which perform the *key service* functionalities, while $G = R - C$ is composed by the generic mesh routers. At least t out of $n = |C|$ core nodes are tamper resistant. A field in the certificate is used to specify the subset to which the node has been assigned by the network operator.
- Synchronization of all mesh routers is needed; this can be achieved using for example the NTP protocol.
- Time is divided into *sessions*, in which all nodes use the same key list generated from the same secret. Therefore, each *session* defines the maximum validity time of the secret, and its duration is equal to the product of the maximum number of keys generated with the secret and the validity time of a generic key.

As for the channel model, we assume for simplicity a free-space propagation model. Furthermore, the channel gain between any two nodes is assumed to be the same in both directions and stationary. As a consequence, all nodes in the network communicate using symmetric channels; therefore, we do not consider in this paper security issues deriving from asymmetric channels.

The basic notation used in the rest of the paper is reported in Table I.

Table I. Basic notation used in the paper

S	Session secret
t_s	Starting validity time of the session secret
K_i^{-1}	Private key of node i
K_i	Public key of node i
K_k^{-1}	Private key of the *key service*
K_k	Public key of the *key service*
$Cert_i$	Certificate of node i
$E_{K_i}(m)$	Message m is encrypted using the public key of node i
$F_{K_i^{-1}}(m)$	Node i 's digital signature of message m
C	Set of core mesh routers, whose cardinality is n
G	Set of generic mesh routers
R	Set of all mesh routers: R=$C \cup G$

5.2. Secret Key Distribution

As illustrated in Section 3, threshold cryptography schemes provide a method to produce digital signatures through the collaboration of a predetermined number of parties.

In the DSA-Mesh architecture, the *key service* consists of n special mesh routers (the *core* routers), which collaboratively generate the new session secret and provide it to the other backbone nodes (the *generic* mesh routers). In particular, all nodes know the *key service* public key K_k, whereas its private key K_k^{-1}, used to sign the secret of each session, is split into n pieces $\left\{K_{k1}^{-1}, K_{k2}^{-1}, ..., K_{kn}^{-1}\right\}$ and each share is assigned to one of the n core nodes.

After having deployed the t tamper resistant nodes, the choice of how many core mesh routers to install (n) presents the following trade-off: increasing n improves the system redundancy and reliability, but it may also increase the overhead and the delay due to the message exchange required by the secret agreement protocol. Furthermore, at most $n = 2t - 1$ core mesh routers can be deployed, since in this case an adversary can recover only up to $t - 1$ pieces of the *key service* private key, and it can therefore generate at most $t - 1$ partial signatures, which do not provide a valid signature for any other node.

The distribution of the *key service* requires the design of a new protocol to deliver the session secret. More specifically, we propose a proactive request protocol, detailed in Protocol 1, which is performed by all generic mesh routers; each of these nodes has to obtain at least t different responses to authenticate the secret used in the current or successive sessions.

A generic mesh router $i \in G$, after entering in the radio range of a mesh router already connected to the wireless backbone, broadcasts its first request to the entire network to obtain the secret S used in the current session by the other routers that form the backbone, and the time when it was generated, t_s. Such request is signed with its private key, K_i^{-1}, and contains the certificate that binds the node identity with its public key K_i.

Each core node j that receives the request from node i, after verifying the authenticity of the certificate, sends back the session secret and the

DOI: 10.1002/sec

Protocol 1 Distributed Proactive Request Protocol

1: A generic node $i \in G$ broadcasts the following message to every core node $j \in C$:
$i \rightarrow j : M_i = (i, Cert_i), F_{K_i^{-1}}(M_i)$

2: Core node j checks the signature on message M_i with public key K_i

3: If the signature is correct, j sends the following response:
$j \rightarrow i : j, M = E_{K_i}(S, t_s), F_{K_{kj}^{-1}}(M)$

4: Node i waits to collect t different responses, and combines the partial signatures to obtain the *key service* signature

5: Node j checks the signature with the *key service* public key K_k

6: If the signature is correct, i decrypts M to obtain the next session secret S and its timestamp t_s

timestamp encrypted with the public key of i, i.e. $E_{K_i}(S, t_s)$, and signs the message with its partial secret of the *key service* private key, K_{kj}^{-1}.

Node i, after receiving at least t different responses, combines them and verifies the digital signature of the message using the *key service* public key, K_k. If the digital signature of the message is correct, then i decrypts the message and obtains the secret S, used by all mesh routers to create the key sequence of the current session. Finally, node i, based on the instant at which it joins the backbone, computes the key currently used to protect the wireless backbone and its remaining validity time ($T_{1,i}$), according to the following equation, where $timeout$ represents the key validity time:

$$
\begin{aligned}
& r_i = \left\lfloor \frac{t_{now} - t_s}{timeout} \right\rfloor + 1 \\
& T_{1,i} = r_i \cdot timeout - (t_{now} - t_s) \\
& \begin{cases} key(r_i, S) = hash(S) & \text{if } r_i = 1 \\ key(r_i, S) = hash(key(r_i - 1, S)) & \text{if } r_i > 1 \end{cases}
\end{aligned} \tag{4}
$$

To enhance the security of the whole system, the argument of the hash function can be obtained by concatenating the secret S and the timestamp t_s with a pre-shared secret known by all nodes.

It is important that each node obtains the secret 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 core nodes (due, for example, to slow links or high number of hops from the core nodes). In fact, if the request is sent when the current session is about to expire, the first nodes that receive the response will cut off the others when they enable the new key.

The key index value that triggers the proactive request protocol can be set equal to the difference between the maximum number of keys in a session and a correction factor, which can be estimated based on parameters such as the network load, the maximum distance that separates the generic node from the core nodes, and the previous time to obtain the responses.

In our architecture, such correction factor (c_i) is computed based on the time necessary to receive the last response from the core nodes (Δt_i), which is estimated according to Equation (5), where $t_{p,i}$ is the time when the first or the previous proactive secret request was sent by node i, and $t_{r,i,j}$ is the time when the corresponding secret response was received from core node j. Note that these values are computed locally and are different for each generic node. So, if a node takes a time (Δt_i in Equation (5)) greater than $timeout$ to receive at least one of the responses from the core nodes, it must perform the next proactive request before setting the last key (otherwise, it will not have enough time to obtain all the necessary responses).

$$
\begin{aligned}
& \Delta t_i = max\left\{t_{r,i,j} - t_{p,i}\right\} \\
& \begin{cases} c_i = \left\lceil \frac{\Delta t_i - timeout}{timeout} \right\rceil & \text{if } \Delta t_i \geq timeout \\ c_i = 0 & \text{if } \Delta t_i < timeout \end{cases}
\end{aligned} \tag{5}
$$

Note that the distributed proactive request protocol is executed by each mesh router when the index of the installed key equals the difference between the maximum number of keys in a session and the correction factor c_i.

Figure 2 shows an example network with the message exchanges performed between generic and core nodes. A (5,3) threshold scheme is adopted, that is, there are $t = 3$ out of $n = 5$ tamper resistant core nodes; black and white circles represent core and generic nodes, respectively.

For the sake of clarity, we draw only the messages necessary to compute the signature, which are represented by solid arrows for requests (from generic to core nodes), and by dashed arrows for core node responses. Moreover, we suppose that generic nodes 1 and 2 join the network for the first time.

Figure 3 shows the message exchanges between such generic nodes and the core mesh routers. As

DOI: 10.1002/sec

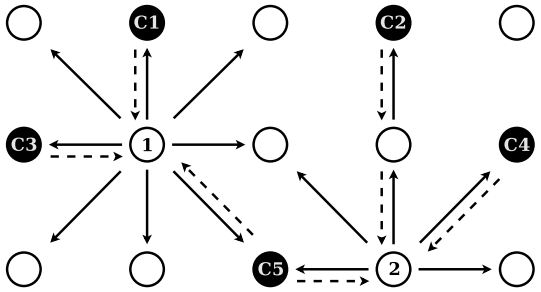


Fig. 2. Distributed Proactive Request Protocol. A (5,3) threshold scheme is adopted in this example WMN, where black and white circles represent, respectively, core and generic nodes. Solid lines represent requests, while dashed lines represent core node responses.

illustrated in Figure 3(a), node 1 performs the second proactive request when the last key is set (i.e., the correction factor is equal to 0), since in this example the $timeout$ value is long enough to obtain all the necessary responses before the session expiration. On the other hand, as illustrated in Figure 3(b), node 2 performs the second proactive request when the third key is set (i.e., the correction factor is here equal to 1), since during the first message exchange it has taken a time greater than $timeout$ to get the three responses.

We observe that, according to our assumptions (Section 5.1), a field of each node's certificate specifies its role. It is therefore unnecessary to announce core nodes: a generic mesh router can discover which nodes act as core nodes using the information contained in the reply message in step 3 of Protocol 1. Even if an adversary is able to compromise a mesh router, obtaining its certificate and gaining access to the backbone network, it cannot impersonate or masquerade other network entities, since the certification authority that releases the credentials is trusted.

Finally, note that an adversary can only compromise at most the $t-1$ core nodes that are not tamper resistant. However, if a compromised core node provides a false session secret, mesh routers can detect its misbehavior by comparing such secret with that received from tamper resistant nodes. In case of mismatch, an alarm can be generated in order to trigger an intrusion detection system that will revoke the certificate of the compromised node.

5.3. Session Secret Agreement Protocol

DSA-Mesh requires that each node of the core set agrees with all the other core nodes on the same secret. To this aim, we have designed a protocol, based on Shamir's no-key protocol, that permits to distribute the session secret to all other core nodes. The message exchange is detailed in Protocol 2.

Protocol 2 Session Secret Agreement Protocol

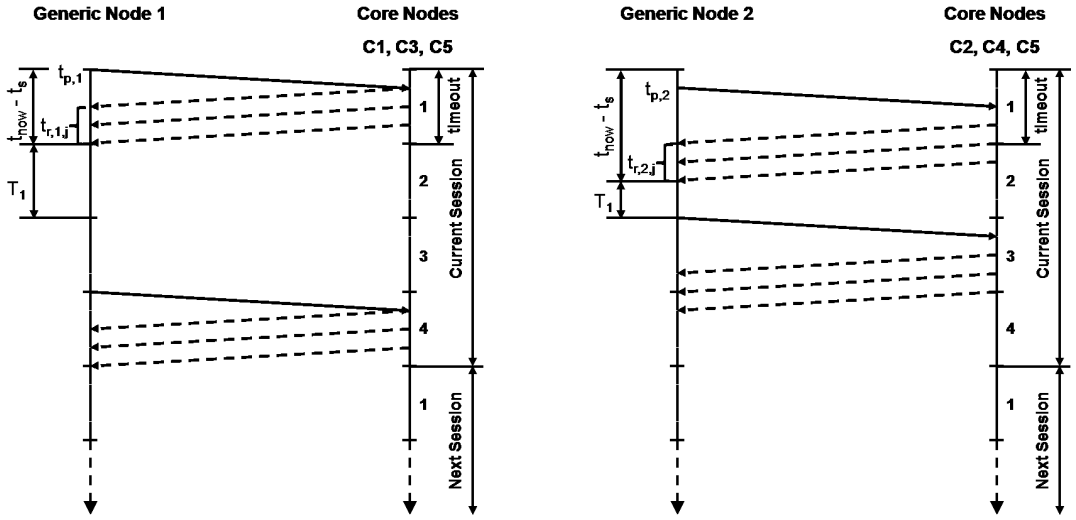
1: The core nodes select a peer as master of the session. Let A be that node.
2: A selects a random prime p and a random number a coprime to $p-1$
3: A broadcasts the following message:
$A \rightarrow B : M = (A, S^a \ mod\, p, p, Cert_A), F_{K_A^{-1}}(M)$
4: Each core node B that receives the previous message verifies the authenticity of the provided certificate and the integrity of the message (by verifying the digital signature).
5: B selects a random number b coprime to $p-1$ and sends the following message:
$B \rightarrow A : M = (B, S^{ab} \ mod\, p, Cert_B), F_{K_B^{-1}}(M)$
6: The master verifies the authenticity of the provided certificate and the integrity of the message
7: If the source of the received message is a core node, A computes $(S^{ab})^{[a^{-1} \ mod\,(p-1)]} \ mod\, p$ and sends the following message to B:
$A \rightarrow B : M = (A, S^b \ mod\, p, Cert_A), F_{K_A^{-1}}(M)$
8: Each core node B that receives the previous message obtains the secret for the successive session, computing $(S^b)^{[b^{-1} \ mod\,(p-1)]} \ mod\, p = S$

Periodically, a specific core node A generates a random secret S for the new session, selects a large prime p such that the computation of the discrete logarithm modulo p is infeasible and a random number a coprime to $p-1$ such that $1 \leq a \leq p-2$. Then, it sends to all the other core nodes a message composed by its identity, the secret S raised (modulo p) to a, the value of the prime p, its certificate, and the signature computed on all the above parameters with its private key.

When a core node receives the message, it verifies the authenticity of the certificate checking the signature of the certification authority that released it, and further checks the signature of the message in order to verify its integrity and authenticity. Then, it chooses a random number b coprime to $p-1$ such that $1 \leq b \leq p-2$, and computes $S^{ab} \ mod\, p$, including in the reply to node A the obtained result along with its identity as well as its certificate. Finally, it signs the message with its private key.

Node A waits for the $n-1$ replies, and after verifying the message integrity and the sender's identity (since only the core nodes can participate

DOI: 10.1002/sec



(a) Correction factor = 0

(b) Correction factor = 1

Fig. 3. Example of correction factor computation in the network topology illustrated in Figure 2

to the secret agreement protocol) it sends the last message to all core nodes. For each received message, A computes $(S^{ab})^{[a^{-1} \ mod \ (p-1)]} \ mod \ p = S^b \ mod \ p$, and inserts in the reply the obtained result along with its identity, its certificate and the signature of all these parameters.

When a core node receives the last message from node A, it extracts the secret for the successive session computing $(S^b)^{[b^{-1} \ mod \ (p-1)]} \ mod \ p = S$, and prepares the partial proactive response signing the value S with its share of the *key service* private key.

In order to secure the whole *key service* system against *node compromise attack*, only one of the t tamper resistant core nodes can become master of the secret agreement protocol. In fact, in this case an adversary cannot distribute fake session secrets to all other core nodes, and therefore it cannot generate a valid signature for generic nodes.

Note that the selection of the successive master (the first step in Protocol 2) can be performed by the master of the previous session by adding a "*next_master*" field to the message M sent in step 7; this field simply indicates whether the destination node (belonging to the tamper resistant set of t core nodes) will act as master of the successive session or not. In this way, the network operator must select only the core node that will act as master in the first session. The master selection process could be performed using, for example, a uniform probability distribution over all tamper resistant core nodes. However, more sophisticated election procedures could be devised.

5.4. Comments and Enhancements

Client Security

To achieve the highest possible level of transparency, the access mechanism to the wireless mesh network can be 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 DSA-Mesh is implemented to protect the backbone links.

DOI: 10.1002/sec

Core Nodes Placement

The deployment of the n core nodes is a key element for the performance of the DSA-Mesh architecture. In fact, each generic mesh node should be sufficiently close to at least t core nodes, in order to collect in the shortest possible time the partial signatures necessary to obtain the key service signature.

Therefore, the optimal placement of core nodes can be formulated as a variation of the *t-neighbor n-center* problem, where t is the number of core nodes to which each generic mesh router wants to be close, while n is the total number of core mesh routers installed in the network. Such problem is known to be NP-hard; however, several polynomial time approximation algorithms that achieve a constant approximation factor have been proposed to solve different versions of this problem [33].

Synchronization Issues

As we stated in the assumptions (Section 5.1), synchronization of all mesh routers is a requirement for our architecture. However, in the preliminary tests performed in [34] we measured a synchronization difference among all nodes always shorter than a few milliseconds. Therefore, a tolerance can be introduced to consider clock drifts. 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 a few seconds before the expiration of the current one, and maintaining the previous key a further few seconds after its expiration.

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 core nodes exchange different cryptographic information for each possible channel, whereas each generic mesh router requests and obtains the cryptographic information that is related only to the wireless channels on which its interfaces are set.

6. Numerical Results

In this Section we present the numerical results obtained testing the proposed security framework in different network scenarios, using Network Simulator. To this aim, we implemented both the protocols defined by DSA-Mesh (namely the *Session Secret Agreement* and the *Proactive Request* Protocols) as an agent entity operating at the routing level, in order to make our architecture independent from the routing protocol.

We compare DSA-Mesh to a static key approach, which consists in securing the WMN with a fixed key; such scheme provides a bound to the performance that can be obtained by the proposed scheme, in terms of achievable throughput, while it is, obviously, a weak solution from the security point of view.

We further compare DSA-Mesh to MobiSEC (see Section 4), since this latter represents a centralized architecture that provides an excellent security solution, but it suffers from a single point of failure and cannot be applied, unlike DSA-Mesh, to large scale self-configuring mesh networks.

We first consider as performance figure the *goodput* of a long-lived TCP connection established between the two farthest nodes of the analyzed topology, to evaluate the effect of the two protocols and the key renewal procedure in the worst case scenario. The goodput is defined as the bandwidth actually used for successful transmission of useful data (payload). Packets are routed over shortest-paths, which are statically computed for all node pairs; this is meant to reduce the overhead due to routing protocols, thus allowing us to evaluate more precisely the effect of our security architecture on the network performance.

Then, we measure the *delay* necessary to perform the proactive request protocol, which provides an indication of the protocol responsiveness. In particular, we analyze the average and maximum delays experienced by all generic nodes to receive the response from the Key Server (in MobiSEC) or the last response from the core nodes (in DSA-Mesh).

Finally, we quantify the reliability improvement achieved by DSA-Mesh with respect to centralized security architectures, developing a mathematical analysis of the availability of our proposed system.

We measure such performance figures using the Multi-Hop, Grid and Random network scenarios illustrated in Figures 4(a), 4(b) and 4(c), respectively. More specifically, the Multi-Hop topology of Figure 4(a) is composed of N nodes, with $N \in \{10, 15\}$, whereas the Grid topology of Figure 4(b) contains 30 nodes placed over a $2000\ m\ \times\ 2000\ m$ area. In the Random topology of Figure 4(c), 30 nodes are uniformly distributed at random over a $800\ m\ \times\ 800\ m$ area.

DOI: 10.1002/sec

The transmission range of a sample node, equal to 250 m, is illustrated in the figures, and black circles represent the core node positions; the carrier sensing range is 550 m when using the highest power level (these are the default values of ns v.2). Even if such settings decrease the spatial reuse, they reduce the hidden terminal problem and therefore the collision probability when two generic nodes perform the proactive request at the same time. The maximum channel capacity is 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.

For the sake of clarity, all the parameters used in our simulations are listed in Table II.

Table II. Parameters used in the simulations.

UDP Packet size	1000 byte
TCP Packet size	1500 byte
Data channel rate	54 Mbit/s
Reception Threshold	-64 dBm
Carrier Sense Threshold	-82 dBm
Capture Threshold	10 dB

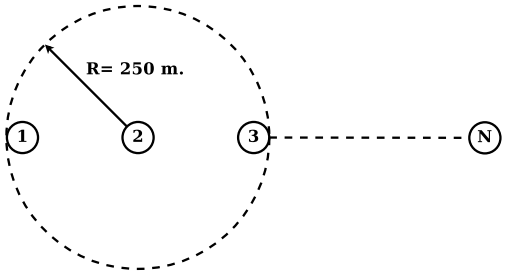
We analyze the proposed protocols, varying the key validity time (the $timeout$ parameter described in the previous Section), and the session duration in order to have 4 keys for each session, since this is the number of hardware registers commonly provided by commercial wireless boards to install the cryptographic keys.

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 simulation time on which we evaluated the performance was equal to 3000 seconds.

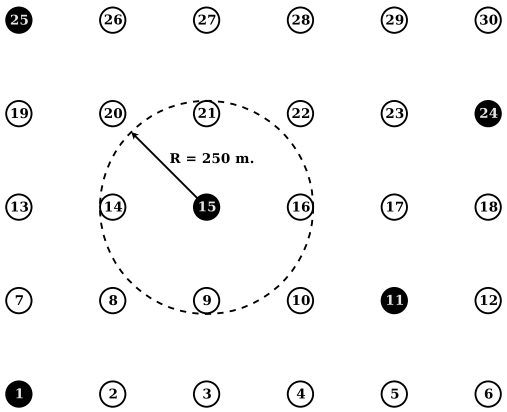
6.1. Goodput Performance

We first measured the *average goodput* of a TCP connection between two nodes. In particular, in the Multi-Hop topology the TCP connection was established between nodes 1 and N ($N \in 10, 15$), whereas in the Grid network it was established between the bottom left and top right nodes (nodes 1 and 30 of Figure 4(b)). In the Random topology, TCP packets are routed on the path $\{24-17-18-13-8-16\}$, illustrated with a bold line in Figure 4(c).

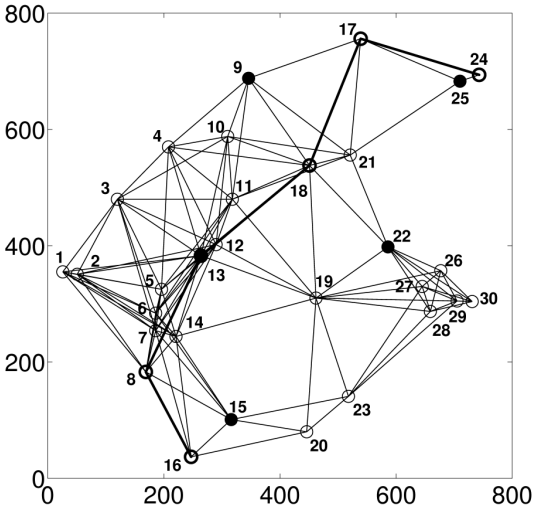
In the simulations that involve the centralized architecture, we placed the Key Server at the center of the analyzed topology. More specifically, nodes 5 and 7 play this role in the Multi-Hop topology with N



(a) Multi-Hop topology



(b) Grid topology



(c) Random topology

Fig. 4. Network topologies used to measure the performance of the proposed security architecture: (a) Multi-Hop topology, composed of N nodes, with $N \in \{10, 15\}$, (b) Grid topology, composed of 30 nodes placed in a 2000 m $\times$ 2000 m area, and (c) Random topology, composed of 30 nodes uniformly positioned at random over a 800 m $\times$ 800 m area.

DOI: 10.1002/sec

equal to 10 and 15, respectively. In the Grid topology, this role is assumed by node 15, while in the Random topology the Key Server is installed at node 19.

TCP NewReno was used for TCP sources, and receivers implemented the Delayed ACKs algorithm. The Maximum Segment Size was equal to 1500 bytes.

Figure 5 shows the TCP goodput obtained in all the considered topologies as a function of the key timeout. As expected, the performance degrades when the key validity time is short, since the greater amount of messages exchanged by the mesh routers reduces the available network bandwidth, and the greater number of key switchings increases the packet drop rate. We observe that this latter problem can be mitigated by setting a little tolerance on the key validity time (as discussed in Section 5.4), since with such technique both early and late nodes can properly decrypt the received frames.

Note that, for a key timeout of 60 seconds, the performance of DSA-Mesh practically overlaps that of the centralized architecture, and it is very close to the bound provided by the static key approach. Hence, DSA-Mesh represents a very effective alternative in scenarios where network operation and control must be provided in a fully distributed, autonomic and serverless fashion, like in wireless community networks.

We further notice that in [34] we measured experimentally that setting the key validity time to 60 seconds turns out to be a conservative and very effective choice, since with such setting we observed that no cryptanalytic attack was able to recover the keys used to encrypt the frames, even when our architecture was used with a weak cryptographic mechanism.

6.2. Delay Performance

In order to gauge the responsiveness of DSA-Mesh, we measured the delay necessary to perform the proactive request protocol in different network configurations.

To evaluate the behavior of the proposed architecture in real scenarios, we set up a background UDP data transfer that involves all network links.

For completeness, Table III reports the mesh routers selected to act as core nodes for the different threshold schemes and network topologies adopted in this analysis.

Table III. Set of mesh routers selected as core nodes for the considered threshold schemes and network topologies.

Threshold Scheme	**(3,2)**	**(5,3)**	**(7,4)**
Multi-Hop 10 nodes	3, 6, 9	2, 4, 6, 8, 10	1, 3, 4, 6, 7, 9, 10
Multi-Hop 15 nodes	4, 8, 12	2, 5, 8, 11, 14	2, 4, 6, 8, 10, 12, 14
Grid 30 nodes	2, 17, 21	1, 11, 15, 19, 29	9, 10, 15, 16, 17, 21, 22
Random 30 nodes	13, 15, 22	9, 13, 15, 22, 25	5, 6, 7, 12, 13, 14, 19

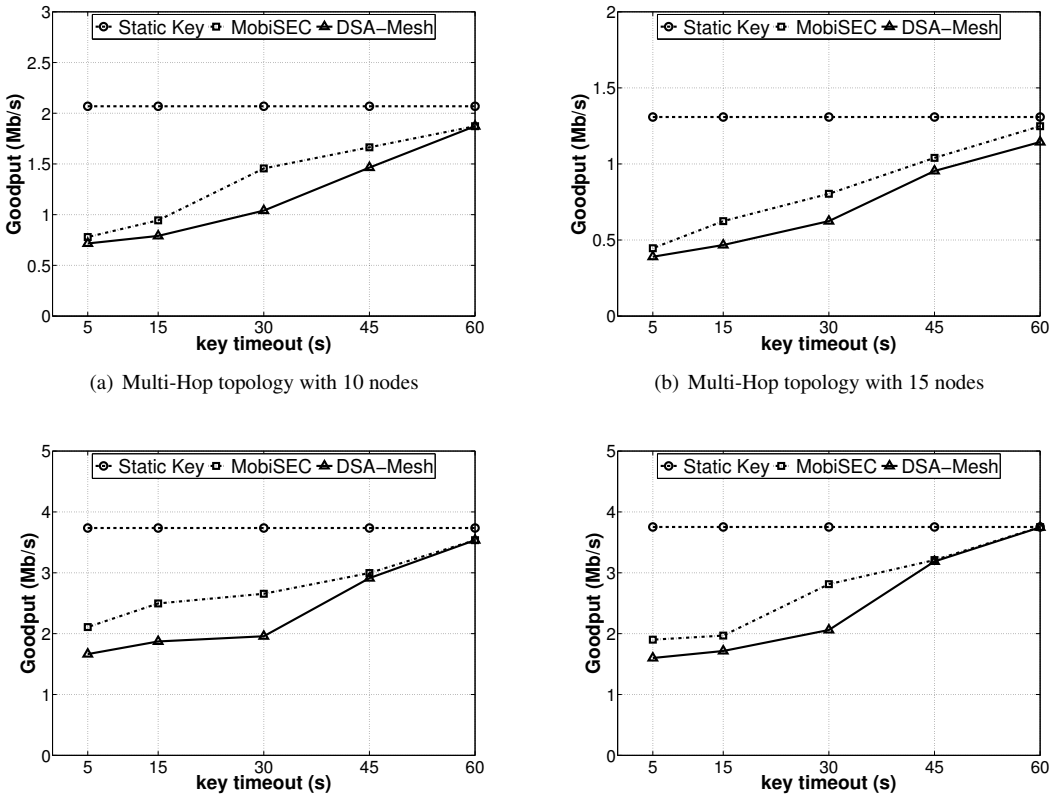
Figure 6 shows the *average delay* (in seconds) measured by all generic nodes as a function of the network load imposed by the UDP traffic, that is, from 100 kbit/s to 1.1 Mbit/s. The lines identified by *"center"* and *"corner"* labels report the results obtained with the centralized security architecture (MobiSEC): the first line corresponds to a network configuration where the Key Server is installed at the center of the analyzed topology (i.e., in nodes 5 and 7 of the Multi-Hop topology with 10 and 15 nodes, respectively, in the central node 15 for the Grid topology and in node 19 for the Random network); on the other hand, the "corner" line reports the results obtained installing the Key Server at the topology border (i.e., in nodes 10 and 15 of the Multi-Hop topologies, in the corner node 30 for the Grid topology and in node 24 for the Random network). The last three curves in Figure 6 show the results obtained by DSA-Mesh using different (n, t) threshold schemes, viz. $(3, 2)$, $(5, 3)$ and $(7, 4)$.

It can be observed that the average delay increases with increasing network load.

In the Multi-Hop topologies the centralized architecture is more responsive than DSA-Mesh, when the Key Server is placed at the center of the network. However, when the Key Server is installed at the topology border (the "corner" curve) the centralized architecture exhibits higher delays, which are larger than those experienced by DSA-Mesh, regardless of the (n, t) scheme deployed. We further observe that, in the Multi-Hop topology with 10 nodes, the $(7, 4)$ threshold scheme performs better than the $(5, 3)$ scheme, and is practically overlapped to the centralized architecture. This is due to the core nodes position used in the simulation. In fact, in the $(7, 4)$ scheme, the three generic nodes are distant at most 3 hops from a core node, whereas in the $(5, 3)$ scheme some responses come from more remote nodes (up to 6 hops).

In the Grid and Random topologies, the performance improvement of DSA-Mesh with respect to the centralized architecture is less evident than in

DOI: 10.1002/sec



(a) Multi-Hop topology with 10 nodes

(b) Multi-Hop topology with 15 nodes

(c) Grid topology with 30 nodes

(d) Random topology with 30 nodes

Fig. 5. Goodput measured in the network topologies of Figure 4(a), the Multi-Hop topology, and Figure 4(b), the Grid network.

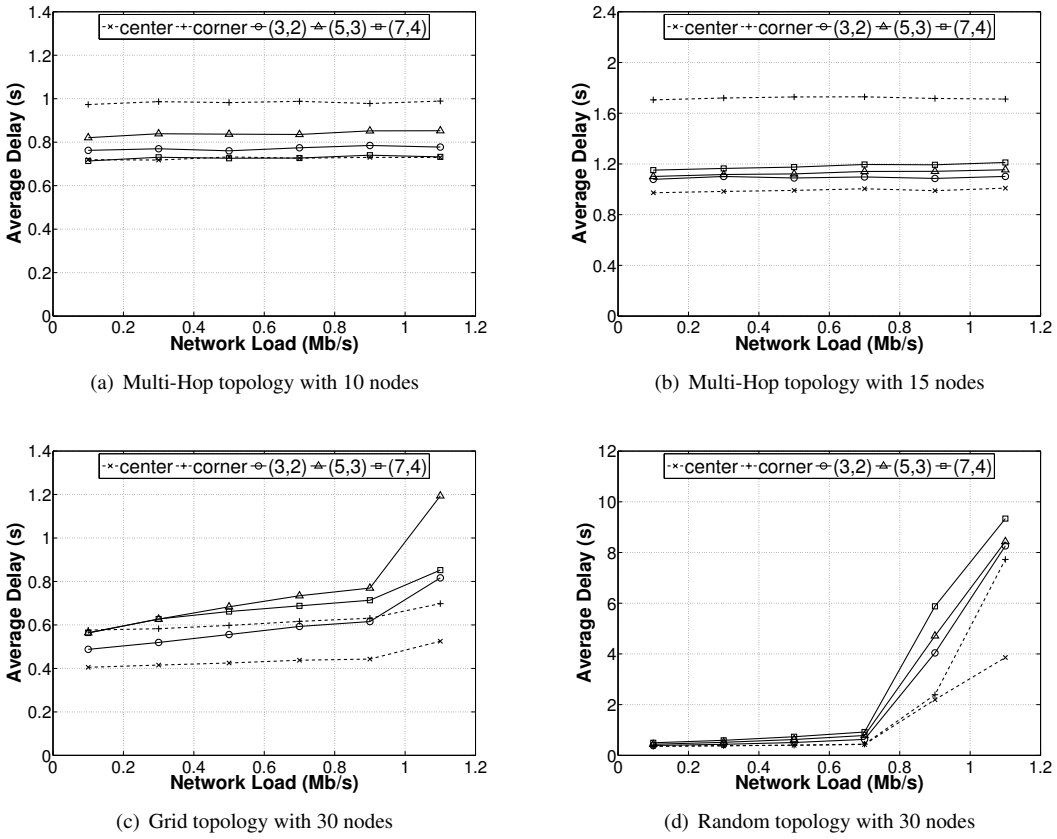
the Multi-Hop scenario, since the network diameters are limited. More specifically, in the centralized architecture, the maximum distance between the mesh routers and the Key Server (5 hops, in the "corner" configuration) still guarantees a slightly shorter delay than waiting for the t replies in the DSA-Mesh scheme. We underline, however, that when the network diameter increases, DSA-Mesh outperforms the centralized architecture, as it is evident in the Multi-Hop topologies.

To provide a more complete comparison, we also measured the *maximum delay*, which provides an indication of the worst-case performance of the DSA-Mesh architecture; the corresponding numerical results are shown in Figure 7. As expected, in the Multi-Hop topologies DSA-Mesh outperforms the centralized architecture when the Key Server is placed at the network border (the "corner" curve); but, unlike the average case, the distributed solution results more responsive than the centralized approach even when the Key Server is installed at the topology center, as it is evident from Figures 7(a) and 7(b).

In the Grid scenario (see Figure 7(c)), due to the limited network diameter, the centralized approach shows a slightly better performance than DSA-Mesh, even though the delay curves are very close for network loads below 1 Mbit/s. A similar behavior can be observed in the Random topology (Figure 7(d)), where the maximum delay experienced by DSA-Mesh well approaches that of the centralized architecture, even for high network loads.

We further observe that, in the grid scenario, the $(7, 4)$ threshold scheme performs better than the $(5, 3)$ scheme (see Figures 6(c) and 7(c)). This is essentially due to the random distribution used to install core nodes, which provides in this case a non-optimal placement. However, as pointed out in Section 5.4, the core nodes' selection problem can be formulated as a variation of the t-neighbor n-center problem, thus permitting to determine optimal deployments of core routers. The investigation of the impact of such optimal choice on the system performance is left as

DOI: 10.1002/sec



(a) Multi-Hop topology with 10 nodes

(b) Multi-Hop topology with 15 nodes

(c) Grid topology with 30 nodes

(d) Random topology with 30 nodes

Fig. 6. Average Delay (in seconds) measured in the Multi-Hop, Grid and Random topologies.

future research issue.

6.3. Reliability Analysis

DSA-Mesh improves the overall network reliability with respect to a single-server centralized approach; in fact, our architecture can tolerate up to $t-1$ core nodes' failures.

Hereafter we illustrate a simple model that permits to quantify the reliability improvement provided by the DSA-Mesh architecture. To this aim, we base our mathematical study on the following classical assumptions [35]:

- a core node has only two states: it is either available or unavailable.
- Different network nodes fail independently leading to repair actions.
- Sufficient resources are available to repair simultaneously any number of failed nodes, restoring them to be as good as new. This is known in the literature as *unlimited repair* [35].
- For any core node, the inter-failure time and the repair time are independent stationary Markovian processes with known mean values: Mean Time To Failure ($MTTF$) and Mean Time To Repair ($MTTR$), respectively.

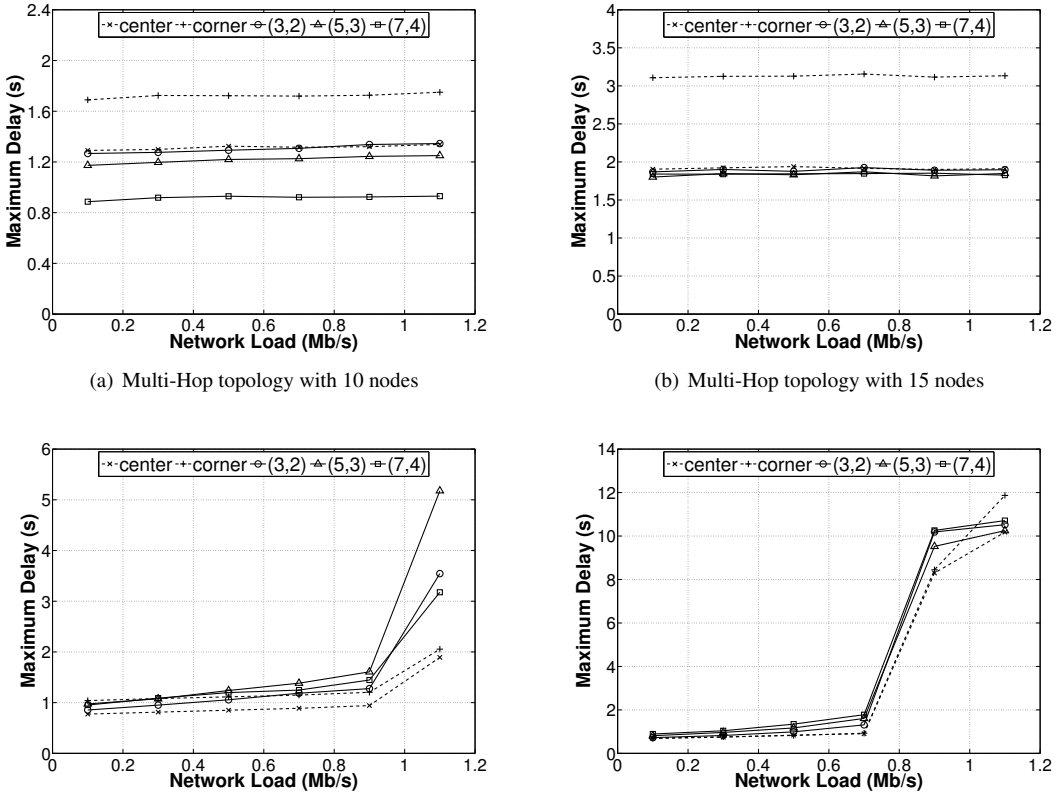
To gain insight into the behavior of the system and according to existing literature [35], we will consider a case of special interest in which all the core nodes have identical failure and recovery rates, equal to λ and μ, respectively. Let us define $\rho = \frac{\lambda}{\mu}$.

The steady-state availability A of a single core node i, viz. the limiting ($\tau \to \infty$) probability of finding it successfully operating at time τ, can be calculated as follows:

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{1/\lambda}{1/\lambda + 1/\mu} = \frac{1}{1+\rho} \tag{6}$$

Furthermore, if all the core nodes failures are statistically independent we have a classical problem of reliability, with $t-1$ out of n redundant resources.

DOI: 10.1002/sec



(a) Multi-Hop topology with 10 nodes

(b) Multi-Hop topology with 15 nodes

(c) Grid topology with 30 nodes

(d) Random topology with 30 nodes

Fig. 7. Maximum Delay (in seconds) measured in the Multi-Hop, Grid and Random topologies.

We can represent the system through the Markov chain illustrated in Figure 8, where state i represents the number of broken core nodes in the network.

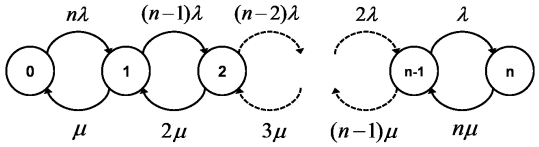


Fig. 8. Markov chain representing the transition diagram of the *key service*. State i represents the number of broken core nodes.

The probability that the *key service* is available is equal to the probability that at most $t-1$ core nodes are down, i.e. $\sum_{i=0}^{t-1} \pi(i)$, where the steady state probability $\pi(i)$ of the Markov chain has the following expression:

$$\pi(i) = \binom{n}{i} (1-A)^i A^{n-i} = \binom{n}{i} \frac{\rho^i}{(1+\rho)^n} \tag{7}$$

Hereafter we compare the security architectures considered in this Section, assuming a $MTTR = 24h$ and $MTTF = 30$, 60 and 90 days. Table IV reports the average system down-time, expressed in hours per year, exhibited by the $(3,2)$, $(5,3)$ and $(7,4)$ threshold schemes, as well as that measured for a single-server centralized solution like MobiSEC (the "Single-Server" column). It can be observed that the overall network reliability is considerably increased with increasing t values, and the total system down-time is dramatically decreased to few minutes per year, especially for the $(7,4)$ scheme.

7. Conclusion

In this paper we proposed DSA-Mesh, a novel distributed security architecture tailored for wireless

DOI: 10.1002/sec

Table IV. Average down-time (expressed in hours per year) exhibited by the considered threshold schemes. $MTTR$ is equal to 24 h, while $MTTF$ is varied from 30 to 90 days. The down-time of a single-server solution, like MobiSEC is further reported for comparison.

Threshold Scheme	**Single-Server**	**(3,2)**	**(5,3)**	**(7,4)**
$MTTF = 30$ days	282.598	26.762	2.803	0.308
$MTTF = 60$ days	143.138	6.986	0.377	0.021
$MTTF = 90$ days	96.272	3.154	0.114	0.004

mesh networks. DSA-Mesh addresses the security problems of the backbone area of WMNs, providing an effective and transparent security solution for end-users and mesh nodes.

We implemented DSA-Mesh in Network Simulator, and tested it in several realistic network scenarios, including large-scale network instances, comparing its performance with that of existing schemes, viz., static key encryption and a centralized security architecture.

Numerical results show that DSA-Mesh offers secure network services to mesh devices with negligible impact on network performance, since it achieves high transmission rates and low latencies. Furthermore, it is intrinsically robust since it does not exhibit a single point of failure, and can be used to deploy automatically and incrementally large wireless mesh networks, thus representing an effective solution for wireless mesh networking.

Acknowledgments

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

References

1. I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Elsevier Computer Networks*, 47(4):445–487, March 2005.
2. R. Bruno, M. Conti, and E. Gregori. Mesh networks: commodity multihop ad hoc networks. *IEEE Communications Magazine*, 43(3):123–131, March 2005.
3. N. Ben Salem and J.-P. Hubaux. Securing wireless mesh networks. *IEEE Wireless Communications*, 13(2):50–55, April 2006.
4. C. Adjih, D. Raffo, and P. Mühlethaler. Attacks against OLSR: Distributed key management for security. In *Proceedings of the 1st OLSR Interop and Workshop*, San Diego, CA, USA, August 2005.
5. W. Stallings. *Cryptography and Network Security, Fourth Edition*. McGraw-Hill, September 2003.
6. C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Mühlethaler, and D. Raffo. Securing the OLSR protocol. In *Proceedings of the 2nd Mediterranean Workshop on Ad-Hoc Networks (Med-Hoc-Net)*, Mahdia, Tunisie, June 2003.
7. D. Raffo, C. Adjih, T. Clausen, and P. Mühlethaler. An advanced signature system for OLSR. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks (SASN'04)*, pages 10–16, Washington DC, USA, October 2004.
8. N. Komninos, D. Vergados, and C. Douligeris. Detecting unauthorized and compromised nodes in mobile ad hoc networks. *Elsevier Ad Hoc Networks*, 5(3):289–298, 2007.
9. H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-securing ad hoc wireless networks. *Proceedings of the 7th International Symposium on Computers and Communications (ISCC 2002)*, pages 567–574, Taormina, Italy, July 2002.
10. N. Milanovic, M. Malek, A. Davidson, and V. Milutinovic. Routing and security in mobile ad hoc networks. *IEEE Computer*, 37(2):61–65, February 2004.
11. L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, November 1999.
12. O. Cheikhrouhou, M. Laurent-Maknavicius, and H. Chaouchi. Security architecture in a multi-hop mesh network. In *Proceedings of the 5th Conference on Security and Network Architectures (SAR 2006)*, Seignosse, France, June 2006.
13. R. Fantacci, L. Maccari, T. Pecorella, and F. Frosali. A secure and performant token-based authentication for infrastructure and mesh 802.1X networks. In *Proceedings of Infocom '06*, Barcelona, Spain, April 2006.
14. P. Antoniadis, B. Le Grand, A. Satsiou, L. Tassiulas, R.L. Aguiar, J.P. Barraca, and S. Sargento. Community building over neighborhood wireless mesh networks. *Technology and Society Magazine, IEEE*, 27(1):48–56, 2008.
15. Vint project u.c. berkeley/lbnl, ns-2 network simulator (ver. 2). *Available at URL: http://www.isi.edu/nsnam/ns/*.
16. F. Martignon, S. Paris, and A. Capone. Design and Implementation of MobiSEC: a Complete Security Architecture for Wireless Mesh Networks. *Elsevier Computer Networks*, 53(12):2192–2207, August 2009.
17. *IEEE Standard 802.11i. Medium Access Control (MAC) security enhancements, amendment 6*. IEEE Computer Society, 2004.
18. *IEEE Standard 802.1X. Port-Based Network Access Control*. IEEE Computer Society, 2004.
19. 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.
20. M. Kassab, A. Belghith, J.-M. Bonnin, and S. Sassi. Fast pre-authentication based on proactive key distribution for 802.11 infrastructure networks. In *Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modeling (WMuNeP'05)*, pages 46–53, Montreal, Quebec, Canada, 2005.
21. A. R. Prasad and H. Wang. Roaming key based fast handover in WLANs. *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'05)*, 3:1570–1576, New Orleans, LA, USA, March 2005.
22. Y. Zhang and Y. Fang. Arsa: An attack-resilient security architecture for multihop wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(10):1916–1928, October 2006.
23. Y. Fu, J. He, R. Wang, and G. Li. Mutual authentication in wireless mesh networks. In *Proceedings of ICC'08*, pages 1690–1694, Beijing, China, May 2008.
24. 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*, Oakland, CA, USA, May 2005.

DOI: 10.1002/sec

25. S. Yi and R. Kravets. Moca: Mobile certificate authority for wireless ad hoc networks. *Proceedings of the 2nd Annual PKI Research Workshop (PKI03)*, pages 612–613, Gaithersburg, MD, USA, April 2003.
26. G. Xu and L. Iftode. Locality driven key management architecture for mobile ad-hoc networks. *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, pages 436–446, Fort Lauderdale, Florida, USA, October 2004.
27. S. Capkun, L. Buttyan, and J.-P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, pages 52–64, 2003.
28. B. Wua, J. Wua, E. B. Fernandeza, M. Ilyasa, and S. Magliveras. Secure and efficient key management in mobile ad hoc networks. *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2005)*, Denver, Colorado, USA, April 2005.
29. J. Kim and S. Bahk. Meca: Distributed certification authority in wireless mesh networks. *Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, pages 267–271, Las Vegas, NV, USA, January 2008.
30. A. Shamir. How to share a secret. *Communications ACM*, 22(11):612–613, 1979.
31. Y. Desmedt and Y. Frankel. Shared generation of authenticators and signatures. *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'91)*, pages 457–469, Santa Barbara, CA, USA, August 1991.
32. T. Dierks and C. Allen. The TLS protocol version 1.0. *RFC 2246*, January 1999.
33. S. Khuller, R. Pless, and Y.J. Sussmann. Fault tolerant k-center problems. *Theoretical Computer Science*, 242(1):237–246, 2000.
34. F. Martignon, S. Paris, and A. Capone. MobiSEC: a Novel Security Architecture for Wireless Mesh Networks. *Proceedings of the 4th ACM symposium on QoS and security for wireless and mobile networks (Q2SWinet '08)*, pages 35–42, Vancouver, British Columbia, Canada, October 2008.
35. J.E.Angus. On computing mtbf for a k-out-of-n:g repairable system. In *IEEE Transactions on Reliability*, volume 37(3), pages 312–313, August 1988.

DOI: 10.1002/sec