FSASD: A Framework for Establishing Security Associations for Sequentially Deployed WMN

André Egners, Hendrik Fabelje and Ulrike Meyer
Research Group IT Security, UMIC Research Center, RWTH Aachen University, Germany

***Abstract*—Wireless Mesh Networks (WMN) mainly consist of an infrastructure of mesh routers (MRs) that are wirelessly interconnected. In many application scenarios these MRs are placed in publicly accessible places and may therefore be compromised by an attacker. Any security framework for WMNs should thus be able to cope with compromised mesh routers. In addition, mesh clients (MCs) are often assumed to be able to route traffic for each other. Such routing MCs, as well as compromised MRs, may try to eavesdrop on and manipulate any type of traffic flowing through them. As a consequence end-to-end protection of all communication in the mesh has to be ensured. Neither the upcoming standard 802.11s nor prior research proposals of security frameworks adequately address this challenge. In addition, many research proposals are incompatible to the upcoming standard therefore only have a slight chance of getting widely used with commercially available devices. In this paper we propose a comprehensive framework for securing wireless mesh networks that is fully compatible to the upcoming 802.11s. The framework enables the efficient establishment of all security associations required for end-to-end protection of the different traffic types in the mesh. In addition, the framework supports secure proactive handovers. We implemented the entire framework in our WMN testbed and present the performance results in this paper.**

***Index Terms*—Wireless Mesh Networks, EAP, Key Management, Security, Bootstrapping, Security Associations, RADIUS.**

I. Introduction

Wireless mesh networks typically form a hierarchy in which infrastructure nodes are wirelessly interconnected. On top of the hierarchy, *Mesh Gateways* (MGs) provide access to other networks, e.g., the Internet. On a second hierarchy level, *Mesh Routers* (MRs), potentially placed in easily accessible areas, route traffic within the WMN. On a third level, *Mesh Clients* (MCs) are connected to the network via MRs. MGs, MRs, and possibly also MCs serves as point of network attachment for other nodes in the WMN. An Authentication, Authorization, and Accounting (AAA) server controls the access to the WMN. As any wireless network, WMNs are vulnerable to external attackers trying to eavesdrop on or manipulate traffic sent over the wireless links or trying to gain unauthorized access to the network. However, the multi-hop nature of wireless mesh networks combined with the potentially exposed placement of MRs and with the fact that MCs may route traffic for other MCs induces additional security challenges for WMNs: compromised MRs and curious or even malicious MCs have to be taken into account. Such MRs or MCs may try to eavesdrop on and manipulate any type of traffic flowing through them. Thus, end-to-end protection of all traffic types has to be ensured. In addition, once identified, there needs to be a mechanism to remove compromised MRs from the network.

The 802.11s draft standard for WMNs does not adequately cope with these security challenges. 802.11s only supports link layer protection of the wireless links within the WMN. As a consequence, any compromised MR and any routing MC has access to the plaintext of all traffic flowing through it, e.g., user traffic from and to the Internet. In addition, all MRs share a single password based on which they authenticate each other when joining the network [1]. Thus, a single compromised MR already has devastating consequences: the attacker – now in possession of the network password – can add more compromised MRs to the network thus assuring that all traffic in the WMN is flowing through a compromised MR. In addition, the use of a network-wide password makes it nearly impossible to manage network access and to remove a compromised MR from the network.

Other previous approaches (e.g., [2] or some of the ones discussed in [3]) support the establishment of some of the required security associations. However, none of these approaches adequately protects against compromised MRs and routing MCs. In addition, most prior approaches (e.g. [4]) are not compatible to the upcoming 802.11s and therefore only have a slight chance of getting widely used with commercially available devices.

In this paper, we address the WMN specific security challenges described above. In particular, we propose a framework that (1) allows for mutual authentication between any node and the AAA server based on any desired type of AAA credentials; (2) supports the removal of any network node (e.g., a compromised MR); (3) solves the problem of bootstrapping security associations required for the end-to-end protection of the different traffic types within a WMN in a highly efficient way; (4) supports end-to-end protection with the help of standardized, already well-scrutinized protocols, namely EAP for node authentication, 802.11i CCMP for link layer protection and IPsec ESP for network layer protection of multi-hop traffic. In addition, our framework supports secure proactive handovers of moving MCs (or MRs) from one point of network attachment to another. Our proposed framework is fully compatible to 802.11s and can easily be realized with commercially available devices.

We implemented the entire framework in our live WMN testbed and evaluated the performance of its components. The results of the evaluations are included in this paper.

978-1-4673-1239-4/12/$31.00

II. Preliminaries

In this section we briefly describe the security architecture IEEE 802.11i for WLANs and a 3-party key transport protocol proposed by Marin-Lopez et al. [5]. Our proposed Framework FSASD makes use of these mechanisms.

A. IEEE 802.11i

IEEE 802.11i [6] allows for mutual authentication between a WLAN client and an Authentication, Authorization and Accounting (AAA) server over an access point with the help of EAP [7]. EAP is an extensible authentication protocol supporting various authentication methods, called EAP-methods. The protocol runs on top of the 802.11 MAC layer between the client and the access point and is encapsulated in RADIUS messages between the access point and the AAA server. During authentication, the AAA server (typically a RADIUS server) and the WLAN client generate two keys. The first one is called *Master Session Key* (MSK) and is sent from the AAA server to the access point encapsulated in a RADIUS message. In the IEEE 802.11i 4-way the client and the access point authenticate each other based on a key derived from the MSK and derive keys for link layer protection between them. The traffic is encrypted and at the same time authenticated using AES with the Counter Mode with Cipher Block Chaining Message Authentication Code Protocol (CCMP). CCMP is a combination of the counter mode (CTR) used during encryption and cipher block chaining MAC (CBC-MAC) for message authentication. The second key is the Extended Master Session Key (EMSK) that is generated for future purposes and is never to leave the AAA server.

B. 3-Party Fast Handoff

The three-party protocol of Marin-Lopez et al. [5] allows two parties A and B, that already share a secret key with a third party S, to establish a shared secret key with each other. The protocol consists of the following messages:

(M1) $A \rightarrow B : A, \{N_A, SEQ_{AS}, B\}_{K_{AS}^{auth}}$
(M2) $B \rightarrow S : B, \{N_B, A_{hash}\}_{K_{BS}^{auth}}, A,$
$\{N_A, SEQ_{AS}, B\}_{K_{AS}^{auth}}$
(M3) $S \rightarrow B : \{N_A, N_B, N_S, A, B\}_{K_{AS}^{auth}},$
$\{N_A, N_B, N_S, A, B, K_{AB}\}_{K_{BS}^{auth}}$
(M4) $B \rightarrow A : \{N_A, N_B, N_S, A, B\}_{K_{AS}^{auth}}$

A initiates the protocol with B by sending $M1$ to B. $M1$ includes its identifier A and a token encrypted by A for S with the symmetric key K_{AS}^{auth} shared between A and S. The token contains a nonce N_A, a sequence number SEQ_{AS} and the identity of B. B relays this message as part of $M2$ to S appending its own identity and a token encrypted with the symmetric key K_{BS}^{auth} shared between B and S. B's token contains a nonce N_B and a hash of the identity of A. In $M3$, S send a token for A (encrypted with K_{AS}^{auth}) and a token for B (encrypted with K_{BS}^{auth}) to B. The token for A contains the identities A and B, and the nonces N_A,N_B, and N_S. The token for B additionally contains the key K_{AB}, which is the key to be shared between A and B. In $M4$, B relays S's token for A to A. As the key K_{AB} is derived from a key K_{AS}^{deriv} shared between A and S and the nonces N_A, N_B, and N_S, the initiator A can derive K_{AB} once it received message $M4$.

For our framework we propose a new variant of this three-party protocol (SectionIII-E). We chose this protocol for two main reasons: First, it has been proposed in the context of handover in WLANs in which devices are authenticated using EAP. This perfectly fits the rest of our framework, which is also EAP-based. Second, the security of the protocol has been formally evaluated using AVISPA [8], a widely accepted formal tool.

III. FSASD - Framework for Establishing Security Associations in Sequential Deployment

A. Network Assumptions

Three types of nodes exist in our WMN, namely Mesh Clients (MC), Mesh Routers (MR), and Mesh Gateways (MG). MGs are responsible for Internet access and routing traffic to other networks. MRs route traffic within the WMN and typically have more than one wireless interface. In addition, MGs, MRs, and MCs may serve as the point of network attachment (NAS) for newly joining MCs or MRs. We assume that the WMN (i.e., all MRs and MGs) is operated by a single operator that sets up the WMN one node after another. MRs may be placed in easily accessible (public) areas. Furthermore, we assume that there is at least one AAA server present in the WMN. This AAA server may, but does not have to be co-located with an MG. Each MG, MR, and MC shares authentication credentials suitable with the AAA server. We further assume that MCs (and possibly also MRs) may move within the network such that a secure and efficient handover procedure is required. Within the WMN, nodes communicate with each other directly on the link layer, as well as on higher layers over several wireless hops for network management, routing, or application purposes. In addition, authentication traffic for newly joining nodes is routed to the AAA server typically over several wireless hops. The same holds for user traffic routed through the WMN to and from the Internet.

B. Attacker Model and Security Requirements

Due to their multi-hop nature, WMNs are particularly vulnerable to active and passive external attackers on the wireless links. These attackers may try to gain unauthorized network access or may try to eavesdrop on or manipulate the traffic in the WMN. Moreover, MRs may be placed in easily accessible areas and can therefore be compromised. Routing MCs can also not fully be trusted. Compromised MRs and routing MCs may try to eavesdrop on and manipulate the traffic flowing through them. As a consequence MC traffic to and from MG has to be protected against compromised MRs and routing MCs. Similarly, authentication traffic between a NAS and the AAA server has to be protected against compromised MRs and routing MCs.

This leads to the following security requirements:

R1 Prevent unauthorized nodes from joining the network
R2 Allow for convenient revocation of compromised nodes

R3 Confidentiality, integrity, and replay protection of each direct (single-hop) wireless link & local broadcast)
R4 Confidentiality, integrity, and replay protection between NAS and AAA
R5 Confidentiality, integrity, and replay protection between MC and MG
R6 Confidentiality, integrity, and replay protection between any two nodes in the WMN wishing to communicate with each other
R7 Fast and secure re-authentication during handover

In order to meet R1 and R2 a protocol for mutual authentication between joining nodes and the AAA server is required, as well as a mechanism to exclude compromised nodes from the network. In order to meet the requirements R3-R7 mechanisms to establish security associations between the respective communicating parties have to be bootstrapped.

C. FSASD Overview

Our framework meets all the requirements listed in Section III-B and in particular solves the problem of bootstrapping the necessary security associations. Our proposal is fully compatible with the 802.11s standard and can easily be implemented on off-the-shelf hardware.

To address R1, we propose to use a key-generating EAP-method for mutual authentication between any joining node N_1 and the AAA server. Revoking compromised nodes (R2) can easily be achieved by revoking the respective AAA credentials. During the EAP authentication two keys are generated at the AAA server and the client, the MSK and the EMSK. As in 802.11i the MSK is used to establish a security association between N_1 and its NAS for (multicast) link layer protection with CCMP (supported by 802.11s devices). In addition, N_1 and its NAS establish a group key GMSK to protect link layer broadcasts. As a consequence, broadcast messages sent through the entire network are hop-by-hop protected on the link layer. The second key - the EMSK - is used as root in a hierarchy of keys illustrated in Figure 1. From the EMSK an IPsec security association (containing an encryption key TEK and an integrity key TIK) is derived. If later on N_1 acts as NAS, these keys are used to protect the authentication traffic between N_1 and the AAA server with IPsec (R4). The two remaining keys PAK and KDK in the key hierarchy are used for authentication and key derivation during bootstrapping of the security associations required to meet R5, R6, and R7. For this bootstrapping we propose the 3-Party Handshake for Sequential Deployment (3PHSD) detailed in SectionIII-E below. During the 3PHSD protocol any already authenticated node A can initiate the establishment of a security association with any other already authenticated node B. The node A and the AAA server derive the key MSK-L1 from the shared KDK and the AAA server securely transfers this key to node B.

D. FSASD Key Derivation

The root of the key hierarchy is the *Extended Master Session Key* (EMSK), a key of at least 64 bytes, that must be exported by any key-generating EAP method [7]. The keys that are derived from the EMSK are cryptographically separated. In the following we describe the key derivation process and length of the derived keys.

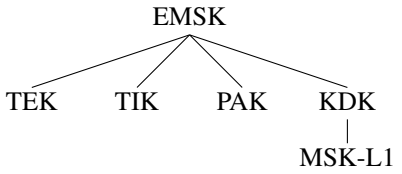

Fig. 1. The *Key Hierarchy* derived from the EMSK.

We use the key derivation function PRF+ specified in RFC 5996 [9], which can be based on any keyed cryptographic hash function. We use HMAC-SHA-256 as default. In addition to the EMSK, PRF+ takes a string indicating the key type, a salt, and the length of the output as input and generates cryptographically independent key material of the desired length, i.e., `KEY=PRF+(EMSK|Name|0x00|Salt|Length)`. If the required key length is unknown at the time of key derivation, the length of the key can be set equal to the length of the EMSK. The TEK (Traffic Encryption Key) and TIK (Traffic Integrity Key) are both 256 bits long. The PAK (Peer Authentication Key), the KDK (Key Derivation Key) and the MSK-L1's (Master Session Key Level 1) are all 64 bytes long.

All keys in the key hierarchy can be configured with a lifetime. The lifetime of each key is ultimately limited by the lifetime of the key it has been derived from, i.e., TEK, TIK, KDK, and PAK must be replaced if the EMSK is refreshed. The EMSK is refreshed only during a full EAP authentication. As the MSK-L1 is derived from the KDK, it may have a lifetime shorter than the lifetime of the EMSK if KDK's lifetime is shorter.

E. 3PHSD - 3-Party Handshake for Sequential Deployment

The goal of 3PHSD is to allow any two already authenticated nodes A and B participating in the WMN to establish a security association with each other (R6). In particular, 3PHSD can be used to set up an IPsec security association between MC and MG (to meet R5) or to setup a link layer security association for CCMP between a moving MC (or MR) and its new NAS during handover. In accordance with Section III-C we use the following notations:

- A, B, S : Identity of Peer A, Peer B, and AAA Server S
- PAK_{AS} : Peer Authentication Key between A and S
- *MSK-L1* : Resulting pairwise key between A and B
- $\{x\}_{k1}$: x encrypted and authenticated by key $k1$
- N_A, N_B, N_S : Nonce of A, B, and S
- t_A : Timestamps of A
- $\{N_A, t_A, B\}_{PAK_{AS}}$: Token 1
- $\{N_A, N_B, N_S, A, B\}_{PAK_{AS}}$: Token 2

A 3PHSD protocol run consists of four messages:

(M1) $A \rightarrow B : A, \{N_A, t_A, B\}_{PAK_{AS}}$
(M2) $B \rightarrow S : A, \{N_A, t_A, B\}_{PAK_{AS}}, N_B$
(M3) $S \rightarrow B : \{N_A, N_B, N_S, A, B\}_{PAK_{AS}}, MSK\text{-}L1$
(M4) $B \rightarrow A : \{N_A, N_B, N_S, A, B\}_{PAK_{AS}}$

Message $M1$ contains the identity of Peer A and Token 1 authenticated and encrypted by the PAK shared between Peer A and S. Token 1 contains a nonce of Peer A, a timestamp, and the identity of Peer B. In Message $M2$, Peer B sends $M1$ and a nonce N_B to S. As B is already authenticated, the communication between B and S is protected by IPsec (cf. Section III-F). Server S can authenticate and decrypt Token 1 based on PAK_{AS}. Using PRF+ with the KDK shared between Peer A and S as key input and the nonces and identities of peers A and B as salt, S now derives the *MSK-L1*, which will be the shared secret of Peer A and Peer B. In message $M3$, S directly sends the *MSK-L1* to Peer B, along with Token 2 authenticated and encrypted by PAK_{AS}. Peer B is now in possession of the *MSK-L1*. The *MSK-L1* is not sent in plain, but protected by the IPsec connection between Peer B and Server S. Message $M4$ is sent from Peer B to Peer A and contains Token 2 which B has received in message $M3$. Peer A can now authenticate and decrypt Token 2 and use its contents to generate the *MSK-L1* using PRF+ using the KDK as key input and the nonces and identifiers of A and B as salt.

After both Peer A and Peer B are in possession of the *MSK-L1*, they can use it as a basis to establish a security association. If 3PHSD is used during association, e.g., in a handover scenario, the *MSK-L1* can be used during the 802.11i 4-way handshake.

F. Network Deployment

Our framework assumes a sequentially deployed WMN, i.e., nodes are added to the network while some are already present. In this section we describe how the different types of nodes, i.e., *Gateways*, *Mesh Routers*, and *Mesh Clients* are deployed.

1) Mesh Gateway Deployment: In the gateway deployment phase, the AAA server and the MG are set up. We differentiate two scenarios, namely the (1) MG being co-located with the AAA server and (2) the opposite case. In the first case, setting up security associations between the MG and the AAA server is obsolete as they are co-located. In the second case, the MG is authenticated using EAP, thus generating the key hierarchy of FSASD. In particular, this bootstraps an IPsec security association between the MG and the AAA server.

2) Mesh Router Deployment: A newly deployed MR connects to the WMN via some already deployed NAS, and is authenticated to the network using EAP. Once the new MR, in the following MR_1, is authenticated, the keys defined in our key hierarchy (cf. Section III-C) are present at both MR_1 and the AAA server. In particular, MSK_{MR_1} is generated at both MR_1 and the AAA, and $EMSK_{MR_1}$ along with the derived keys, i.e., TEK_{MR_1}, TIK_{MR_1}, PAK_{MR_1}, and KDK_{MR_1}. The PMK derived from the MSK_{MR_1} is used in the 802.11i 4-way handshake between MR_1 and the NAS it is associating to. Once the 4-way handshake succeeded, link layer encryption and integrity protection with CCMP are enabled between MR_1 and the NAS. Next, the multi-hop connection from MR_1 to the AAA server is secured by bootstrapping IPsec with the TEK_{MR_1} and the TIK_{MR_1} for encryption and integrity protection. Any other device (MR_2 or some MC) connecting to MR_1 acting as NAS will benefit from this IPsec connection, as the authentication traffic generated by EAP will be secured from MR_1 to AAA.

3) Mesh Client Deployment: The client deployment is similar to the mesh router deployment, as clients authenticate to the network using EAP as well. The NAS which the client connects to is already connected to the network and therefore IPsec between the NAS and the AAA server, is already enabled. As a result, EAP traffic of the client connecting to the network will be protected between the NAS and the AAA server. After the client has successfully associated to the network, it generates the key hierarchy based on the EMSK.

Using 3PHSD, MCs can now bootstrap a security association for an IPsec *tunnel* connection to the MG. Thereby, all traffic generated by MC and destined to the Internet and other networks is protected throughout the WMN and cannot even be deciphered or manipulated by corrupted MRs in the WMN. Note that an additional IKE handshake is unnecessary as 3PHSD (cf. Section III-E) yields a shared secret key that can directly be used on the IPsec tunnel connection.

G. Pro-active Handover

Figure 2 shows the use of 3PHSD for securing the handover of an MC from MR1 to MR2. 3PHSD allows to pro-actively establish a fresh MSK-L1 between MC and MR2 before handover. This MSK-L1 can then be used in the 802.11i 4-way handshake between the MC and MR2. This makes any full EAP authentication or EAP re-authentication obsolete.

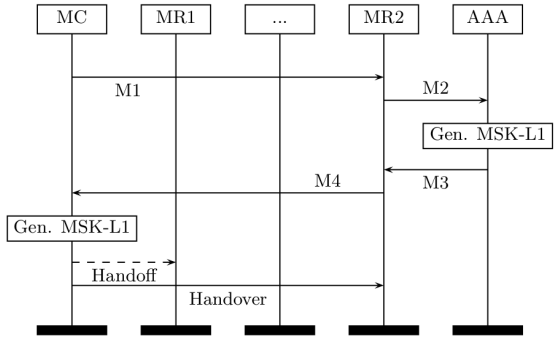

Fig. 2. Pro-active Handover using 3PHSD

In order to minimize the disconnection time, the MC initiates the 3PHSD with MR2 via MR1 before transferring its connection from MR1 to MR2. MC sends the first 3PHSD message $M1$ (cf. Section III-E) to MR2 using SCTP (or UDP). If MR2 wishes to communicate with the MC, MR2 sends message $M2$ to the AAA server. Message $M3$ contains material for the MC to generate the MSK-L1, as well as the MSK-L1 for MR2. $M2$ and $M3$ are protected by the IPsec SA between the MR2 and the AAA server. Message $M4$ is sent from MR2 to the MC on top of SCTP (or UDP), using the IP address and port $M1$ was sent from. Note that MR1 does not learn the fresh MSK-L1 between MC and MR2. The MC can now check whether the received data has been created by the AAA server based on the nonces and the PAK it

shares with the AAA server, extract the nonces and derive the new MSK-L1. The MSK-L1 can then be used as a basis for the 802.11i 4-way handshake, as such, key confirmation is implicitly achieved.

Using 3PHSD for handover takes significantly less time than a full EAP authentication (cf. Section V), therefore also the loss of connectivity is reduced. As the MC is not associated to MR2 before starting 3PHSD, the protocol messages have to be sent via an alternative EAP lower layer, which itself effectively runs on top of IP. This results in the fact that the MC needs to know IP and port, as well as the lower layer protocol to transport EAP. Discovering 3PHSD-enabled MRs could be possible by using 802.11 [10] vendor-specific element being broadcast in 802.11 beacon frames. Alternatively, the MC could acquire a list of MRs, e.g., distributed by the AAA server during authentication, or any in-network service.

Comm. Pattern	SA bootstrapped by	Secured by
single-hop (R3) MR↔ MC, MG, MR	EAP based on AAA credentials	CCMP using keys derived from MSK
single/multi-hop (R4) NAS↔AAA	EAP based on AAA credentials	IPsec using TEK and TIK
multi-hop (R5,R6) MC↔ MC, MG, MR	3PHSD based on PAK	IPsec using keys derived from MSK-L1, derived from KDK
Broadcast (R3) MC, MG, MR →*	EAP based on AAA credentials	CCMP using GMSK derived from MSK
handover (R7) MC↔new NAS	3PHSD based on PAK	CCMP using keys derived from MSK-L1, derived from KDK

TABLE I
PROTECTED COMMUNICATION PATTERNS IN WMNS.

IV. SECURITY ANALYSIS

A. Analysis of FSASD

Table I summarizes the security features and mechanisms FSASD provides and how these relate to the security requirements R3-R7 introduced in Section III-B. Recall that R1 (mutual authentication between any joining node and the AAA server) is met by using an adequate EAP-method. R2 (revocation of compromised nodes) is enabled by revoking the AAA credentials of a node.

Note that FSASD never reveals any keying material to any MR or routing MC except the keys deployed by these nodes themselves. In particular, FSASD does not leak any keys to compromised MRs or curious/malicious routing MCs. All keys transported from the AAA server to a node in the WMN (e.g. MSK-L1 in 3PHSD and PMK during EAP authentication) are protected with IPsec.

A compromise of session keys does not compromise any longer-term keys. All lower-level keys in the proposed key hierarchy, are derived from a higher-level key with the help of a cryptographic hash function (HMAC-SHA-256). Therefore it is not possible to compute any higher-level key from a compromised lower-level key [11]. Compromise of session keys does also not compromise future or past session keys. This is due to the fact that upon any EAP authentication a fresh EMSK is generated such that all keys generated from the EMSK are also fresh.

B. Analysis of 3PHSD

In this section we briefly discuss the security properties of 3PHSD. When the MSK-L1 is sent from the AAA server to B its confidentiality and integrity is protected by the IPsec ESP connection between the AAA server and B. $M2$ and $M3$ are replay protected by IPsec ESP. The AAA server S can detect a replay of Token 1 based on t_A. A can detect replay of Token 2 based on the previously committed nonce N_A. During 3PHSD A and B are authenticated by the AAA server S: B indirectly via the IPsec connection and A by PAK_{AS}. B will always receive the same MSK-L1 from the AAA server S, A will compute, since they are both generated from the same nonces and identities. A compromise of the MSK-L1 does not allow to compute any other shared keys, since it is derived from the KDK using PRF+. Thus, 3PHSD meets the same security goals a the 3-party protocol originally proposed by Marin-Lopez et al. [5] with one exception: token replay can only be detected by the party for which the token is destined and not by the party that forwards it to its intended destination. This, however, only delays the detection of replays for a small amount of time.

We also performed a formal analysis of 3PHSD using AVISPA [8]. It proves that the following specified security goals are achieved: confidentiality of the sent MSK-L1, authentication of the messages between A and S, as well as B and S, S will only send the MSK-L1 to B if B and A are correctly authenticated, Token 1 (destined for S) is replay protected, Token 2 (destined for A) is replay protected, and both $M2$ and $M3$ are replay protected as well.

Note that *3PHSD* differs from the original protocol of Marin-Lopez [5] in the following way: The sequence numbers are replaced by timestamps as we assume loose time synchronization between all network nodes. In M2 of the original protocol, Peer B sends its nonce N_B and a hash of Peer A's identity in an authenticated and encrypted token to S. In our 3PHSD this encryption and the hash are not necessary as the connection between B and S is encrypted and authenticated by IPsec. Similarly, the transfer of the key *MSK-L1* from server S to Peer B in message M3 of our 3PHSD does not have to be explicitly encrypted as the communication between Peer B and S is protected by IPsec. In addition to reducing the overall message size, we thus reduced the number of cryptographic operations, i.e., one less hash operation and two less encryptions in 3PHSD without sacrificing security (see Section IV).

V. PERFORMANCE EVALUATION

In this section we evaluate the performance of our implementation of FSASD. In particular, we evaluate the overhead introduced by FSASD by using IPsec to protect the RADIUS communication between a NAS and the AAA server during

EAP authentication of a newly joining MR or MC. In addition, we evaluate the performance of 3PHSD using two different EAP lower layers, namely UDP [12] and SCTP [13].

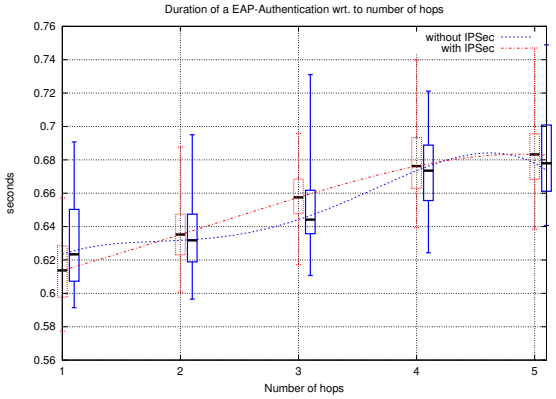

Fig. 3. EAP-TTLS/PAP Authentication Time wrt. hops and IPsec Usage

A. WMN Testbed Setup

Our WMN testbed was set up and evaluated with respect to security and performance using PC Engines ALIX system boards. All devices run on *Voyage-Linux*, which is a Debian Squeeze based embedded Linux distribution. Linux 2.6.38.2 is built in order to support the correct 5 GHz channels and the new ath9k wireless drivers. Each device has a 500 MHz AMD Geode CPU, 256 Megabytes of RAM and two Atheros AR5008 wireless controllers. The first wireless card allows an MR to connect to another device (NAS) of the WMN, while the second card can be used to distribute connectivity, i.e., act as NAS to other MRs or MCs.

When a device boots, it will automatically try to connect to the next MR. As WMNs are *self-healing*, a new connection will automatically be established if a connection is lost. The testbed uses the *batman-adv* routing protocol which will automatically adapt to new network topologies.

B. Performance of EAP Authentication

Figure 3 shows the time required for EAP authentication for different numbers of hops between the NAS and the AAA server. We measured the authentication time with IPsec between NAS and AAA server enabled (red) and disabled (blue). The boxes in the figure represent the lower and upper quartile. The median is marked by the black bar. Each measurement is labeled with the median and minimum and maximum are marked by the whisker-bars.

Our experimental results confirm that the duration of an EAP authentication increases with the distance between NAS and AAA server. In addition, protecting the authentication traffic between NAS and AAA server by IPsec does not significantly increase the overall authentication time.

C. Performance of 3PHSD

We measured the time required for 3PHSD dependent on the number of hops between the 3PHSD peers and the AAA server. We implemented 3PHSD as an EAP method using two different EAP lower layers: SCTP and UDP to transport EAP messages through the WMN. Both of these transport protocols are simple packet-oriented protocols and not stream-oriented like TCP. This allows to efficiently encapsulate the 3PHSD messages in SCTP or UDP messages. The time was measured by the peer initiating the 3PHSD starting from the sending of the first message, over computing and displaying the new MSK until receiving the fourth message.

The tuples on the horizontal axis of Figure 4 represent the distances between Peer A and Peer B, as well as the distances between Peer B and the Server S. For example, *2-3* means that the distance between Peer A and Peer B is two hops, and the distance between Peer B and the AAA server is three hops.

The different variations in distance have been chosen to realistically map the usage scenarios of 3PHSD, i.e., (1) handover and (2) establishing security associations between MC and MG. In the first scenario (1), Peer A represents an MC which wants to initiate a handover to another NAS, i.e., Peer B. The distance from the MC to the destination NAS during the handover can be expected to be shorter than the distance from the NAS to the Server S.

In the second scenario (2), Peer A represents an MC which wants to bootstrap a security association for an IPsec tunnel from MC to MG, i.e., Peer B is equivalent to the MG. The distance from the MC to the MG can be expected to be longer than the distance between MG and the AAA server. MCs will typically connect from the edge of the network.

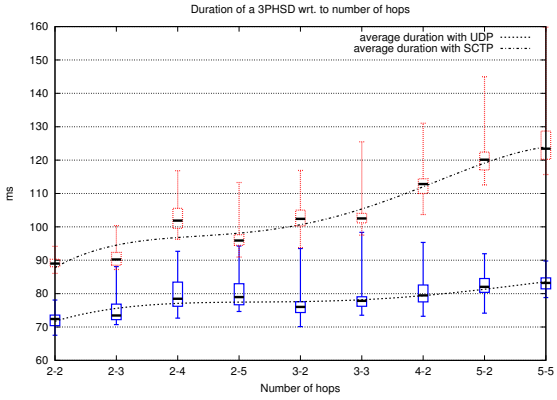

Fig. 4. Authentication Time with 3PHSD Using SCTP (red) and UDP (blue).

It can be seen that the overall distance from Peer A via Peer B to Server S has the biggest impact on the duration of the 3PHSD protocol run. However, larger distances between Peer A and Peer B seem to have a greater impact on the duration than the distance between Peer B and Server S. Using 3PHSD with UDP is faster due to the fact that UDP does not implement a handshake like SCTP does. Also, UDP does not address reliable delivery of messages.

Compared to the EAP authentication time measured in the testbed, 3PHSD is faster regardless of using UDP or SCTP as a lower layer for the EAP messages. 3PHSD along with the SCTP lower layer and two hops requires about 85% less time

than a full EAP-TTLS/PAP authentication run. As seen in this section, EAP-TTLS/PAP authentication requires about 600 ms. With a distance of five hops between Peer A and Peer B, and Peer B and Server S, 3PHSD still requires 82% less time then a full EAP-TTLS/PAP authentication run. Using UDP as a lower layer for 3PHSD outperforms EAP-TTLS/PAP by more than 91%, and even still 89% for the five hop scenario.

VI. Related Work

In [3] Egners et al. identify which communication patterns have to be protected in a wireless mesh network and argue whether integrity protection and confidentiality or integrity protection only is required for these patterns.

The authors also present a comprehensive study of research proposals on key management for WMN including [14], [2], [15], [16] and the upcoming standard [17] and conclude that none of the existing proposals adequately addresses bootstrapping security associations for all of the identified relevant communication patterns in a WMN. Here we solve the problem of bootstrapping the security associations required to protect the suggested patterns in [3] (Table I, left).

Another weakness of previously suggested key management frameworks (e.g., [15]) as well as industrial solutions such as [18] is that they are based on proprietary or at least non-standardized protocols and are often not compatible with the upcoming IEEE 802.11s. In this paper we close this gap and suggest a framework which is fully compatible with IEEE 802.11s and is based on well-scrutinized standardized protocols such as EAP, IPsec and IEEE 802.11i.

In the context of WLANs with a wired infrastructure many proposals for securing handover procedures exist (e.g. HOKEY [19], CAPWAP [20], and the IEEE 802.11r [21] extension, to mention just the standardized ones). HOKEY and CAPWAP both require interaction with the AAA server during handover and the AAA server transfers keying material to the destination access point. Neither HOKEY nor CAPWAP was designed with wireless multi-hop networks in mind. As a consequence both approaches leak keys to intermediate MRs or routing MCs when applied directly and without further protection to WMNs.

In 802.11r, the keys for the link layer protection between the moving MC and the destination access point are derived from the keys shared between MC and the source access point and transferred from the source to the destination access point. This key transfer needs to be protected with the help of a security association. The problem of bootstrapping this SA other than manually is not solved in 802.11r rendering it unsuitable for multi-hop wireless networks such as WMN.

Our handover solution is secure against compromised MRs and routing MCs and is more efficient than using a full EAP authentication with the AAA server when associating with the new NAS.

VII. Conclusion

In this paper we presented FSASD, a novel framework for establishing security associations in sequentially deployed WMNs. Our proposal overcomes the bootstrapping problem of many other key management proposals for WMNs including the current draft standard 802.11s. As opposed to the draft standard our proposal reduces the influence of compromised mesh routers on the over all security of the mesh to a minimum and protects mesh clients from curious and malicious routing MCs. In addition, our framework enables secure and efficient handovers of mesh clients and mesh routers based on the new 3PHSD protocol. Our framework is compliant to 802.11s and is based on well-scrutinized security protocols for link layer and network layer protection as well as authentication during network access. Our performance analysis shows that the performance penalty for the added security features our framework provides is very low such that our framework is well-suited for direct practical use.

Acknowledgments

This work has been supported by the UMIC Research Centre, RWTH Aachen University. Also thanks to Tobias Jarmuzek who was a great help in running and developing the testbed.

References

[1] D. Harkins, "Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks," *SENSORCOMM '08*.
[2] F. Martignon, S. Paris, and A. Capone, "MobiSEC: A Novel Security Architecture for Wireless Mesh Networks," in *Q2SWinet*, 2008.
[3] Andre Egners, Ulrike Meyer, "Wireless Mesh Network Security - State of Affairs," *SICK'10, 35th IEEE LCN*.
[4] Y. Zhang, "ARSA: An Attack-Resilient Security Architecture for Multihop Wireless Mesh Networks," *Selected Areas in Communications'06*.
[5] Lopez et al., "Secure three-party key distribution protocol for fast network access in EAP-based wireless networks," *Computer Networks'10*.
[6] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 6: Medium Access Control (MAC) Security Enhancements (IEEE Std 802.11i-2004)," Jul. 2004.
[7] B. Aboba et al., "Extensible Authentication Protocol (EAP) Key Management Framework," RFC 5247, 2004.
[8] L. Vigan, "Automated Security Protocol Analysis With the AVISPA Tool," *Electr. Notes Theor. Comput. Sci.*, 2006.
[9] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)," RFC 5996 (Proposed Standard), IETF.
[10] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, June 1999.
[11] P. Rogaway and T. Shrimpton, "Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance," *FSE*, 2004.
[12] J. Postel, "User Datagram Protocol," RFC 768, IETF, Aug. 1980.
[13] L. Ong and J. Yoakum, "An Introduction to the Stream Control Transmission Protocol (SCTP)," RFC 3286, IETF, May 2002.
[14] O. Cheikhrouhou, M. Laurent-Maknavicius, and H. Chaouchi, "Security Architecture in a multi-hop Mesh Networks," in *SAR*, 2006.
[15] Zhang and Fang, "A Secure Authentication and Billing Architecture for Wireless Mesh Networks," in *Wireless Networks*, 2007.
[16] K. Ren, "A Sophisticated Privacy-Enhanced Yet Accountable Security Framework for Metropolitan Wireless Mesh Networks," in *ICDCS'08*.
[17] "IEEE 802.11s Task Group, Amendment: ESS Mesh Networking, D3.0."
[18] Motorola, "MOTOMESH 1.2 - Guidelines for Network Planning'06."
[19] T. Clancy et al., "Handover Key Management and Re-Authentication Problem Statement", RFC 5169."
[20] P. Calhoun, "Control And Provisioning of Wireless Access Points (CAPWAP) Access Controller DHCP Option," RFC 5417.
[21] IEEE, "802.11: Amendment 2: 802.11r Fast Basic Service Set (BSS)."