A Comprehensive Security Architecture for Multi-Operator Wireless Mesh Networks

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

André Egers, Dipl.-Inform.
aus Siegen, Deutschland

Berichter: Prof. Dr. Ulrike Meyer
           Prof. Dr. Dieter Gollmann
Tag der mündlichen Prüfung 30.09.2014

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.
To my grandparents Alfred and Johanna.
Acknowledgments

During the process of obtaining the degree I interacted with a diverse set of people, all of which deserve my thanks.

First and foremost I thank Prof. Dr. Ulrike Meyer for agreeing to be my thesis advisor and giving me that chance to work with her. Besides setting a great example I very much enjoyed the easy and candid atmosphere during our meetings. All around the institute the pleasant work environment was a great help.

Some of my thanks go to Andriy Panchenko without whom I would very likely not have chosen to pursue the path of obtaining a doctoral degree. Experiencing his thesis work first hand and our lengthy discussions about this particular career path were a good indicator helping in my decision.

I hereby also thank my students Hendrik Fabelje, Patrick Herrmann, and Tobias Jarmuzek which I have advised during their theses in my field of research. Special thanks goes to Tobias who maintained our WMN testbed with eternal tenacity. All of their results contributed positively to my research in terms of publications.

Additionally, I thank all my colleagues who have been working at the institute over the years. We had great fun, ranging from “group building” to our daily rage sessions during the lunch breaks, be in persona in IRC-a.

I also thank all of my friends which have endured me raging or “nerding” about some or all parts of my thesis work, or work in general. It was great to have a place where I could discharge my ramblings.

Last but not least I thank my parents and my grandparents for their endless support and faith in me. It surely has not been easy, especially in the beginning of my studies of Computer Science. Special gratitude goes to my grandpa Alfred Müller who unfortunately cannot witness my graduation any more.
I have never thought, for my part, that man’s freedom consists in his being able to do whatever he wills, but that he should not, by any human power, be forced to do what is against his will. – Jean-Jacques Rousseau
A Comprehensive Security Framework for Wireless Mesh Networks

Abstract: Wireless Mesh Networks (WMNs) represent one of the key technologies that are used to cope with the increasing demand of ubiquitous connectivity and the accompanying hunger for bandwidth. Due to their wireless nature WMNs are very flexible in their deployment. However, flexibility often comes at the price of security. WMNs have to be secured against external, as well as against internal attackers. Special attention has to be paid to all communication patterns in the network, since otherwise no comprehensive security can be achieved.

This thesis proposes a comprehensive security architecture for WMNs that extends standardized mechanisms such as the Extensible Authentication Protocol (EAP), the Remote Dial-in User Service (RADIUS), IEEE 802.11i, and the Internet Protocol Security (IPsec) suite. We compose an architecture that allows to bootstrap security associations based on an extensible key hierarchy. Besides enabling secure communication between authenticated devices, our architecture is generalized to support multi-operator scenarios. This also includes completely new concepts such as mixed-networks in which network operators cooperate in running a converged network. Our comprehensive security architecture is augmented by handover protocols that enable network clients, but also the network infrastructure, to hand over from one point of network attachment to the next. The complete architecture has also been evaluated using a live, custom-built WMN testbed based on off-the-shelf hardware. This underlines the feasibility and practicality of the work put forth in this thesis.

Keywords: Wireless Mesh Networks, Security, Multi-Operator, Handover, Roaming, Testbed.
# Contents

1 Introduction 11
   1.1 Contributions ......................................................... 15
   1.2 Roadmap ............................................................... 20

I Fundamentals 25

2 Ingredients 29
   2.1 Extensible Authentication Protocol ................................. 29
      2.1.1 EAP-Tunneled-TLS ................................................. 31
   2.2 Remote Dial-in User Service ......................................... 31
      2.2.1 Attacking RADIUS ............................................... 32
   2.3 Wireless LAN ........................................................ 33
   2.4 Internet Protocol Security ........................................... 38
   2.5 The pwrcall RPC Framework ......................................... 40
   2.6 Summary ............................................................. 42

3 Security Challenges of Wireless Mesh Networks 47
   3.1 Security Architectures and Mechanisms for WMNs ................... 47
   3.2 Wireless Mesh Networks .............................................. 49
      3.2.1 Node Types ..................................................... 49
      3.2.2 Communication Patterns ....................................... 50
   3.3 Attacker Model ...................................................... 51
   3.4 Scenarios, Characteristics & Challenges .............................. 52
      3.4.1 Single-Operator Wireless Mesh Networks ....................... 52
      3.4.2 Multi-Operator Wireless Mesh Networks ....................... 53
      3.4.3 Heterogeneity of the Wireless Links ............................ 54
      3.4.4 Routing Clients ............................................... 54
      3.4.5 Mobility ....................................................... 55
   3.5 General Security Requirements ...................................... 56
   3.6 Security Architectures for WMNs .................................... 58
      3.6.1 IEEE 802.11s - Mesh Networking ............................... 59
      3.6.2 Attack Resilient Security Architecture for WMNs ............... 62
      3.6.3 MobiSEC ........................................................ 63
      3.6.4 A PANA-based Security Architecture for WMNs ................. 65
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Wireless Mesh Network Architecture [AW05]</td>
<td>12</td>
</tr>
<tr>
<td>2.1</td>
<td>The EAP communication stack.</td>
<td>29</td>
</tr>
<tr>
<td>2.2</td>
<td>A generalized EAP message flow.</td>
<td>30</td>
</tr>
<tr>
<td>2.3</td>
<td>The format of RADIUS packets.</td>
<td>32</td>
</tr>
<tr>
<td>2.4</td>
<td>Message flow of the EAPOL protocol.</td>
<td>35</td>
</tr>
<tr>
<td>2.5</td>
<td>Message flow of the IEEE 802.11i 4-Way handshake.</td>
<td>36</td>
</tr>
<tr>
<td>2.6</td>
<td>Comparing the IPsec tunnel modes.</td>
<td>39</td>
</tr>
<tr>
<td>2.7</td>
<td>Comparing the IPsec transport modes.</td>
<td>39</td>
</tr>
<tr>
<td>3.1</td>
<td>An exemplary WMN architecture.</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>The message flow of the ECC-based SAE Protocol used in the IEEE 802.11s amendment.</td>
<td>61</td>
</tr>
<tr>
<td>4.1</td>
<td>The exemplary setup of the ITsec testbed.</td>
<td>77</td>
</tr>
<tr>
<td>4.2</td>
<td>An exemplary view of the internals of an ALIX.3D3 board.</td>
<td>78</td>
</tr>
<tr>
<td>4.3</td>
<td>An overview on the architecture of the pwrmesh software.</td>
<td>80</td>
</tr>
<tr>
<td>4.4</td>
<td>An excerpt of the node view of pwrmesh showing the details of meshrouter16 along with a map.</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>An example of the deploy view of pwrmesh showing a selection of functionalities.</td>
<td>82</td>
</tr>
<tr>
<td>4.6</td>
<td>The packet loss ratio in the ITsec testbed</td>
<td>83</td>
</tr>
<tr>
<td>4.7</td>
<td>The average round trip times in two different setups.</td>
<td>84</td>
</tr>
<tr>
<td>4.8</td>
<td>The average throughput of the ITsec testbed</td>
<td>84</td>
</tr>
<tr>
<td>5.1</td>
<td>The messag flow of the 3PFH protocol.</td>
<td>97</td>
</tr>
<tr>
<td>5.2</td>
<td>The Framework for establishing Security Associations for Sequentially Deployed WMN (FSASD) key hierarchy derived from the EMSK.</td>
<td>100</td>
</tr>
<tr>
<td>5.3</td>
<td>The message flow of the 3PHSD protocol.</td>
<td>104</td>
</tr>
<tr>
<td>5.4</td>
<td>Dividing a WMN into logical FAU Sub-Domains using the AAA server as the top level controller.</td>
<td>108</td>
</tr>
<tr>
<td>5.5</td>
<td>The FSASD key hierarchy extended such that FAU support is enabled.</td>
<td>109</td>
</tr>
<tr>
<td>5.6</td>
<td>The security associations in a FAU-enabled WMN. The colors used correspond to Figure 5.5.</td>
<td>110</td>
</tr>
<tr>
<td>5.7</td>
<td>EAP-TTLS/PAP runtime with respect to wireless hops and IPsec usage between the MC and the AAA server.</td>
<td>116</td>
</tr>
</tbody>
</table>
5.8 3PHSD runtime using SCTP (red) and UDP (blue)........ 118

6.1 An abstraction of the multi-operator network model used in this thesis. 127
6.2 The domain specific FSASD key hierarchy for the Operators $\pi$ and $\psi$. 129
6.3 Authentication of $MC^\psi_\pi$ which is roaming in the FN of Operator $\pi$. 131
6.4 Bootstrapping an IPsec SA between the roaming $MC^\psi_\pi$ and $MG^\psi_\psi$ of the MC’s HN. 132
6.5 Bootstrapping an IPsec connection from the roaming $MC^\psi_\pi$ to an MG of the FN. 134
6.6 Authenticating $MR^\psi_\pi$ in the domain of Operator $\pi$. 135
6.7 Authenticating arbitrary devices in the domain of Operator $\pi$. 137

7.1 The general idea of the 3PHSH protocol. ......................... 149
7.2 The 3PHSH protocol message flow. .......................... 149
7.3 The general idea of the NPA protocol. ......................... 150
7.4 The NPA protocol message flow. .......................... 151
7.5 The general idea of the ENPA protocol. ......................... 152
7.6 The ENPA protocol message flow. .......................... 153
7.7 The attribute format used by the ENPA protocol. .......... 153
7.8 The PMK format used by the ENPA protocol. ............... 154
7.9 A snippet of the domain specific FSASD key hierarchy including keys used by the logical FAU domains. 156
7.10 The general purpose evaluation topology that is used for 3PHSH, NPA, and ENPA. 159
7.11 Runtime of 3PHSH comparing TCP and UDP over various distances. 160
7.12 The NPA evaluation topology allowing distance variations to the TMRs. 160
7.13 The NPA star topology focussing on the amount of requested TMRs. 161
7.14 This figure shows the duration after NPA message M2 containing the PMK is received at the TMRs for various distances using TCP and UDP. 161
7.15 This shows the runtime of NPA until the last TMR has received the PMK. 162
7.16 This figure shows the runtime of NPA with regards to the distance between the TMR and the AAA server. Here only one TMR is requested by the MC. 163
7.17 This figures shows the effect of including varying numbers of IP addresses in the ENPA messages received a the Mesh Client (MC). 164
7.18 This figure shows the duration of scanning each channel using IEEE 802.11a and IEEE 802.11b. ................................. 165
## List of Tables

2.1 The IEEE 802.11 beacon frame body. [IEE99] .......................... 37
2.2 The IEEE 802.11 Vendor-Specific Information Element format. ....... 38

3.1 An overview on the security requirements for the different commu-
nication patterns in WMNs. .................................................. 56
3.2 A summary of the characteristics & the security requirements of the
compared architectures. ...................................................... 67
3.3 The mobility functionalities of the compared architectures. ............. 68

4.1 This table compares different WMN testbeds on a general level. ....... 88

5.1 The various key types, key lengths & purposes used in FSASD. ...... 102
5.2 Protected communication patterns in a WMN secured by FSASD. .... 114

7.1 This table summarizes the protocol properties of 3PHSH, NPA, and
ENPA. ........................................................................... 155
7.2 A comparison of the theoretical handover duration for bootstrapp-
ing a *single* handover destination. Note that only NPA and ENPA
are capable of bootstrapping multiple handover destinations in one
protocol run. .................................................................. 159
7.3 This table summarizes the security analysis respective to RFC4962. .. 167
Nowadays, mobile telecommunication networks (e.g. 3G networks) and Wireless LANs (WLANs) co-exist and compete to provide Internet access to mobile users. Over time the number of users with data subscription plans to mobile operators has constantly increased. This has already begun to spark business models aiming to off-load traffic to WLANs. The available bandwidth and speed of a 3G connections is often too slow to enable multimedia applications such as video streaming. This is mostly related to the varying coverage quality in such networks and the large user base in cells. On the one hand operators must heavily invest in expensive hardware to increase network quality. But on the other hand, the immediate Return of Investment (ROI) is far from being clearly defined. Thus, operators usually cap the tariffs in terms of amount of data, e.g., significant speed reduction after consuming 500MB.

IEEE 802.11 WLAN coverage on the other hand is widely available in many places: private individuals operate WLAN Access Points (APs) at their homes, universities and companies provide wireless access to their students and employees, commercial operators provide access at hotels, airports, coffee shops, and other public places. Even to a point of WLAN pollution where the amount of APs decreases the network quality for all users. Usually these networks are either operated by private individuals with a very limited user group, or they are operated by a commercial entity that requires user registration. This results in a collection of connectivity bubbles, each of different quality and administrative domain.

In order to address these issues many so-called community WLAN networks have been initiated, e.g., Freifunk or Fon\(^1\). The point being, a community offers their collective WLAN access to their members. Each member must contribute their private WLAN to be able to profit from access elsewhere. However, this approach has some downsides. Usually private individuals are responsible for the traffic that is transmitted from their network to other networks, e.g., the Internet. As such, they can also be held accountable in a legal sense for any illegitimate or malicious traffic generated by their users. Also, it is non-trivial to enable WLAN access to community members while protecting traffic from being eavesdropped on or manipulated by

\(^1\)http://www.freifunk.net, http://www.fon.com
other community members. In consequence the acceptance of such approaches usually suffers.

In terms of commercial operators it is typically required for users to register separately with each operator. Additionally, adding new infrastructure to a commercial WLAN can be quite cumbersome and expensive as a wired connection from the backbone to the new access point is required and communication protocols may not be standardized.

It would be straightforwardly beneficial for all involved entities to increase the connectivity by leveraging a greater coverage area. A flexible, easy, and on-demand extension of the infrastructure can be achieved by so-called Wireless Mesh Networks (WMNs). Parts of this technology have also been standardized by the IEEE based on WLAN [IEE99] and other networking technologies such as WiMAX [IEE12]. In general, a WMN consists of a hierarchy of nodes that are wirelessly connected with each other (see Figure 1.1). On top of the hierarchy, Mesh Gateways (MGs) provide access to other networks, typically the Internet, and route traffic from the mesh to the Internet and vice versa. On a second hierarchy level, Mesh Routers (MRs) route traffic within the wireless mesh. On a third level, Mesh Clients (MCs) are connected to the network via MRs that serve the clients as points of network attachment. Finally, in WMNs, some MCs are able to route traffic for other MCs and thus act as MCs and MRs simultaneously. Within a wireless mesh network different communication patterns exist. MCs communicate directly with MRs, and with MGs and other MCs; MRs communicate with each other and with MGs. All the communication patterns originating from MCs, MRs, and MGs have to be authenticated and adequately
protected against eavesdropping and manipulation.

Today, WMNs are mostly considered to be operated by a single operator which is also reflected by standardization efforts. For the future, however, we do not only envision WMNs operated by different commercial operators to inter-operate, but also a convergence of commercial and community networks. In such a converged mesh network multiple commercial operators cooperate, such that MCs can obtain network access via MRs operated by any of the commercial operators and yet obtain a single bill (roaming). Private individuals contribute their access points to the converged WMN. In contrast to community networks individuals are compensated for the use of their access points by other participants. MCs can be configured to act as MR for other MC in the converged network and be compensated. Thus, in our envisioned converged mesh networks an oligarchy of operators ranging from commercial operators to private individuals with their mobile equipment will be supported. In addition, we envision such converged networks to support mobility of MCs (seamless handover) as well as mobility of infrastructure components. Seamless mobility for MCs is crucial for real-time applications such voice or video conferencing, and video streaming. Supporting MR mobility allows, e.g., for easy re-connection on relocation of a private individual contributing its WLAN AP, or on setup and tear-down of APs contributed by commercial or public entities in case of temporary events.

In order for all participants to accept and profit from the envisioned converged mesh, it is crucial to ensure that it does not lead to financial losses to the involved commercial and private entities, and that it meets the functional requirements even in the presence of malicious and selfish MCs, MRs or MGs. Within this thesis we develop new security mechanisms that address security for WMNs as whole. We propose mechanisms for bootstrapping security associations, multi-operator networks including infrastructure sharing, roaming, and handover services. Finally, to demonstrate their practicality, all developed mechanisms have been implemented and evaluated using current off-the-shelf hardware.
1.1 Contributions

This thesis addresses the security issues of WMNs on a general level by proposing a comprehensive multi-operator enabled security architecture including bootstrapping, key management, and handover capabilities. In the following we summarize our main contributions.

At first, we contribute significantly to analyzing the security issues of WMNs on a general level in Chapter 3. While there is consensus on the importance of the main security challenges of WMNs, prior work has not yet led to generally accepted security requirements. In particular, existing proposals for security architectures for WMNs tend not to address WMNs as a whole, but rather concentrate on one or two specific characteristics and design security mechanisms that meet the security requirements for these specific characteristics. We step back to classify and characterize WMNs as a whole. This high-level vantage point helps us to define general security requirements and identify unique challenges to meet these requirements with respect to the characteristics of WMNs. We then use this framework of requirements and characteristics to evaluate state of the art proposals ranging from standardization efforts of the IEEE to results from academia.


In recent years many testbeds for WMNs have been implemented to testing and evaluating different aspects of WMNs, however, none of these have been designed with testing and evaluating security mechanisms for WMNs in mind. In Chapter 4 we share from the experience of designing a testbed dedicated to test and evaluate security protocols in a realistic setting. We detail the hardware and software setup of our testbed, the management tools we developed to facilitate maintenance of our testbed, along with several pitfalls we encountered during the setup. This testbed is used in all subsequent chapters to evaluate the developed security mechanisms. The testbed is also continued to be used for further research activities. As such, our testbed significantly contributes to enabling practical research and feasibility evaluations.

André Eghners, Patrick Herrmann, Tobias Jarmuzek and Ulrike Meyer: *Experiences*
A number of security architectures have been proposed for WMNs, however, none is a comprehensive solution. We propose a comprehensive security framework based on open standards and well scrutinized security protocols in Chapter 5 solving the well known bootstrapping problem. In particular, our framework allows for mutual authentication between any node and the AAA server based on any desired type of AAA credentials; supports the removal of any network node (e.g., a compromised MR); 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; supports end-to-end protection with the help of standardized, well-scrutinized protocols, namely EAP for node authentication, 802.11i CCMP for link layer protection and IPsec ESP for network layer multi-hop traffic and supports secure proactive handovers of moving MCs (or MRs) from one point of network attachment to another. Additionally, we introduce a new component that allows to split an administrative WMN domain into logical sub-domains. This has the potential to significantly speed up handover protocols.


While previous security research of WMNs mainly focused on single-operator networks, we propose an extension of our comprehensive security architecture for multi-operator WMNs. Our proposal (cf. Chapter 6) allows for a secure deployment of infrastructure components (routers and gateways) as well as MCs. The multi-operator support of our architecture does not only cover MC roaming, but also the deployment of infrastructure components of one operator in the administrative domain of the other operator. Our architecture is the first to support secure infrastructure sharing between operators. Additionally, our solution is based on open standards and protects traffic generated by mesh clients from insider attackers such as compromised MRs, MRs operated by malicious operators, and curious or malicious routing MCs.

André Egners and Ulrike Meyer: *Secure Roaming and Infrastructure Sharing for Multi-Operator WMNs*, 28th ACM SAC 2012, Coimbra, Portugal [EM13]
The main contributions of Chapter 7 are three complementary secure and efficient handover protocols for WMNs that can be used by all devices, including the network infrastructure. One of these protocols allows to pro-actively supply candidate Target Mesh Routers (TMRs) with keying material as part of the initial Extensible Authentication Protocol (EAP) authentication of a device joining the WMN. The other two protocols can be run at any point in time after successful authentication. The first of these protocols allows several TMRs to be pro-actively supplied with keying material, but the device requiring handover cannot be sure that these TMRs have already received the keying material at the end of the protocol run. The second protocol has the advantage that a device initiating the protocol can be certain that the TMR has received the keying material when the protocol finishes. Additionally, we extend the protocols to be usable in a multi-operator contexts. We implemented and evaluated our protocols using our testbed and integrated them into the de-facto standard WLAN software hostapd and wpa_supplicant.

André Egners, Patrick Herrmann and Ulrike Meyer: Secure and Efficient Handover Protocols for WMNs, 14th IEEE WoWMoM 2013, Madrid, Spain [EHM13]

Previous proposals on WMN security mainly focus on mesh networks operated by a single operator and rarely support mobility of MCs with the help of secure roaming and handover procedures. While these approaches protect the communication of MCs against external attackers, they do not take internal attackers into account. In our previous publications we proposed a security architecture for single-operator WMNs, extended this architecture to the multi-operator case to support roaming between operators and secure infrastructure sharing and proposed secure handover procedures within the domain of a single operator. In this paper we merge the different aspects of our prior proposals together to form a comprehensive security architecture for multi-operator WMNs. Our solution is based on open standards and explicitly addresses internal attackers. In addition, we propose proactive handover services between different operators and show how dedicated MRs can take over authentication services in time critical situations such as handover procedures.

After summarizing the contributions of this thesis we continue with the roadmap stating how we detail the individual concepts.
1.2 Roadmap

This thesis continues with Chapter 2 which introduces basic concepts, building blocks and technologies that are of importance for this work. We start by detailing EAP as the standard mechanism for authentication in wireless networks, specifically in IEEE 802.11 WLAN networks. EAP is an extensible protocol framework including authentication methods and message transport. We further discuss the EAP-Tunneled Transport Layer Security (EAP-TTLS) as the authentication method which is subsequently leveraged by our security architecture. EAP-TTLS tunnels an additional inner authentication method inside of an end-to-end encrypted Transport Layer Security (TLS) tunnel from the EAP peer to the EAP server.

After discussing EAP, we continue with the Remote Dial-in User Service (RADIUS) protocol which encapsulates EAP on the network layer between the Network Access Server (NAS) and the Authentication, Authorization and Accounting (AAA) server. The RADIUS/AAA/EAP server is responsible for authentication and authorization of network devices and users. It stores the necessary credentials, generates keys and transports them to the NAS a device is authenticating from. The originally intended setup uses a wired backbone, thus, the traffic between NAS and AAA server cannot be easily eavesdropped on. However, the key transport messages themselves are only barely secured.

We then introduce the IEEE 802.11 WLAN standard and how both, EAP and RADIUS are used to authenticate wireless devices. This also includes the discussion of the 4-way handshake which is used to establish a link layer security association between the Station (STA) and the AP. We also include a section on the IEEE 802.11u [IEE04b] amendment. It intends to improve interworking with external networks, such as the mobile communication networks. However, it also introduces very useful features for network discovery and AP selection. It allows to include so-called vendor specific Information Elements (IEs) in the IEEE 802.11 wireless beacon. This is a particularly interesting feature which is used in this thesis during handover (cf. Chapter 7) and to announce node types used for optimizing the key distribution (cf. Section 5.3).

Chapter 2 closes by elaborating on Internet Protocol Security (IPsec), a mechanism to secure multi-hop communication on the network layer between two endpoints or networks. We detail the most important features of IPsec such as its tunneling and transport modes.

After having set the necessary technical foundations, this thesis continues by discussing the security challenges of wireless mesh networks in Chapter 3. We start
by introducing the general terms of WMNs such as the different architecture and node types. We then turn to the security challenges of WMNs.

Once we have briefly discussed the related work in researching the security challenges in WMNs, we follow by defining an attacker model for the works subsequently presented in this thesis. Attackers are defined with respect to being external and internal, as well as to their respective capabilities.

Next, the security requirements of WMNs are defined by considering all possible communication patterns, e.g., MC to MG communication. All patterns are analyzed with respect to the security requirements ranging from confidentiality to non-repudiation. Additionally, we broaden these requirements by considering specific challenges and scenarios in WMNs, e.g., single- and multi-operator WMNs, heterogeneity of wireless links and mobility of network clients as well as mobility of the network infrastructure.

Subsequent to having the security requirements and communication patterns for WMNs exemplified, we use them as a framework to evaluate a selection of prominent related work which claims to address security of WMNs in a comprehensive way. This includes the IEEE 802.11s mesh networking amendment, as well as proposals from the research community.

In Chapter 4 we introduce a WMN testbed developed for the purpose of the security research put forth in this thesis. We discuss the requirements that we have towards the testbed and detail the overall architecture, the hardware, the software that we use and mechanism for node management. Additionally, we elaborate on pitfalls that we discovered during setting up the testbed and using it for evaluating our research. Besides shortly discussing other WMN testbeds, we also include a general performance evaluation of the testbed, i.e., Round Trip Time (RTT), throughput, latency and also the performance impact of using cryptographic acceleration cards.

Chapter 5 proposes a framework to bootstrap security associations for WMNs. It represents the foundation of the architecture extensions detailed in the subsequent chapters and addresses all the requirements deduced in the previous chapters. By using EAP-based authentication of all nodes in the network, we are able to extend the EAP security framework by leveraging the presence of the Extended Master Session Key (EMSK). This key serves as the root in an extensible key hierarchy which both node and authentication server share after successful authentication. Based on this key hierarchy we bootstrap various security associations, e.g., for the purpose of secure key transport using EAP/RADIUS in a wireless multi-hop domain. Our framework also allows to split a WMN domain into multiple logical sub-domains for the purpose of speeding up key management protocols and handover (cf. Chap-
This extension is similar to the multi-operator architecture we propose in Chapter 6. The key management framework is also thoroughly evaluated with respect to performance and is implemented using our WMN testbed.

After detailing our security architecture in a single-operator context, we extend our concept to a multi-operator scenario in Chapter 6. We use the notion of so called domain specific keys to extend our key hierarchy. By exporting a copy of the key hierarchy, which is specific to the respective domain, all the security associations that are valid in the single operator concept, can also be used in a multi-operator scenario. Additionally, we introduce the previously unknown notion of mixed networks, in which different operators contribute their devices collaboratively forming a single network.

In Chapter 7 we continue by extending our security architecture even further. The context of WMNs and oblivious network coverage implicates the capability of seamless handover from one wireless access point to the next. We propose three proactive handover protocols for WMNs that leverage the security architecture of Chapter 5 and the multi-operator extension presented in Chapter 6. Our protocols allow to bootstrap handover keys, i.e., Pairwise Master Keys (PMKs) used during the IEEE 802.11i 4-way handshake. This allows us to reduce the overall handover delay to a minimal level as the keys are deployed before initiating the actual handover. We also evaluate the handover protocols using our WMN testbed described in Chapter 4.

Last, Chapter 8 and Chapter 9 conclude this thesis by summarizing the prior chapters and providing an outlook to future research in this area.
Part I

Fundamentals
This chapter introduces the basics concepts and techniques that are required for this thesis. After generally introducing the operation of the Extensible Authentication Protocol (EAP) and specifically EAP-Tunneled Transport Layer Security (EAP-TTLS), we continue with the Remote Dial-in User Service (RADIUS) protocol. Next, we detail how the former mechanisms are used with the IEEE 802.11 [IEE99] Wireless LAN (WLAN) standard to facilitate authentication, authorization, and accounting for WLAN devices. Also we shortly introduce the secure Remote Procedure Call (RPC) framework pwrcall. Last, we present Internet Protocol Security (IPsec) as a general purpose mechanism to secure network layer traffic between two hosts or networks.

2.1 Extensible Authentication Protocol

EAP is specified in Request For Comments (RFC) 5247 [ASE08] as an extensible framework supporting multiple different authentication methods. It is specified to be network layer agnostic, thus, the RFC also defines the basic concept of the lower layer. Typically, data link layers such as IEEE (IEEE) 802 are used. EAP is used in IEEE 802.1X [IEE10] for Port-Based Network Access Control. As no transport protocol such as Transport Control Protocol (TCP) is used, the protocol itself must keep state of the ongoing sessions.

Figure 2.1 shows the EAP stack. The primary goal is to authenticate the peer to the authentication server. The peer and the pass-through authenticator, typically referred to as the authenticator, communicate using an EAP lower layer. In order to

![Figure 2.1: The EAP communication stack.](image)
send the EAP messages from the pass-through authenticator to the authentication server, the latter encapsulates the messages in Authentication, Authorization and Accounting (AAA) messages. Communication between the pass-through authenticator and the authentication server is IP-based.

The basic message flow of EAP is depicted in Figure 2.2. First, the authenticator sends an EAP-Request-Identity frame to the peer which responds with an EAP-Response-Identity containing peer’s identity. The authenticator forwards this message to the authentication server. Next, the authentication server and the peer agree upon which EAP authentication method is to be used. If the chosen EAP method is key generating, two keys will be generated by the peer and the authentication server: the Master Session Key (MSK) and the so-called Extended Master Session Key (EMSK). The MSK is sent by the authentication server in the EAP-Success message to the authenticator which may use it for further tasks. In the context of IEEE 802.11i the first 256 bits of the MSK are used as the input key for the 4-way handshake (cf. Section 2.3). This handshake is used to derive keys which are used to protect the communication on the data link layer between the Station (STA) (peer) and the Access Point (AP) (authenticator).

The second key is the EMSK which must neither leave the peer nor the authentication server. It is reserved for future use as specified in RFC 5247 [ASE08]. If an EAP method is key generating it has to ensure that both the peer and the authentication server are mutually authenticated.
2.2. Remote Dial-in User Service

2.1.1 EAP-Tunneled-TLS

EAP-TTLS [PF08] is a key generating EAP authentication method which extends the Transport Layer Security (TLS) [DR08] protocol. It provides authentication of the AAA server using standard TLS procedures. As EAP-TTLS is \textit{tunneled}, it may encapsulate authentication protocols such as the Peer Authentication Protocol (PAP) or another EAP method, called \textit{inner} method for peer authentication. This is for instance important to protect password based authentication methods against eavesdropping or man-in-the-middle attacks.

EAP-TTLS additionally allows to carry so called \textit{Attribute Value Pairs (AVPs)}. Using such AVPs enables to send vendor specific information from the client to the EAP-TTLS server in an end-to-end secured manner. The AVP format of EAP-TTLS is equivalent to the AVPs used by Diameter [PJE+03] which is the logical succession of the RADIUS protocol (cf. Section 2.2). However, while not using Diameter, it is possible to map these AVPs to RADIUS attributes.

2.2 Remote Dial-in User Service

RADIUS is specified in RFC 2865 [RRSW97]. It is a protocol that carries information to perform authentication, authorization, and accounting. In order to keep the protocol and the implementation simple, RADIUS uses the User Datagram Protocol (UDP). This means that RADIUS must handle retransmission of packets itself. In case one server fails, the packet must be resent to another server.

Four types of packets are defined to be passed between RADIUS server and Network Access Server (NAS): Access-Request, Access-Accept, Access-Reject, and Access-Challenge. A NAS acts as the RADIUS client and communicates with the RADIUS server. The packet format is depicted in Figure 2.3 and includes transport attributes that provide specific information such as the NAS-IP-Address (typically an AP) and attributes that allow authentication and authorization, e.g., the User-Name and User-Password attributes. User information is transported within the Access-Request.

Multiple round trips for authentication are possible by having the server send an Access-Challenge and the client answering with an Access-Request. At the end of an authentication exchange, the RADIUS server replies with an Access-Reject or Access-Accept to the NAS. RADIUS server and NAS share a secret which essentially represents an ASCII string. Passwords in RADIUS packets are obfuscated using the shared secret and MD5 [Riv92].

RADIUS packets have an 128 bit authenticator, such that the NAS is able to verify whether it received the packet from the correct RADIUS server. For the Access-Request this value is randomly chosen by the NAS and the reply contains the
respective authenticator (cf. Equation 2.1, “||” denoting concatenation). The Request Authenticator authenticates the packet the server replies to while the other fields of the received packet (Code, ID, Length, and the Attributes) are protected in the same manner. The parameter Secret is shared between the RADIUS server and the client (NAS).

\[
\text{ResponseAuthenticator} = \text{MD5}(\text{Code}||\text{ID}||\text{Length} \\
\quad ||\text{RequestAuthenticator}||\text{Attributes}||\text{Secret})
\] (2.1)

Encapsulation of EAP-Messages in RADIUS messages is based on RADIUS attributes (cf. Figure 2.3) [AC03]. In order to indicate the attribute type, the Type field is used. If the attribute encapsulates an EAP message, the Type field is set to 79. To indicate a vendor-specific attribute the type must be set to 26. The actual data is contained in the Value field following the one byte long Length field which indicates the size of the attribute. The length includes the Type, Length, and the Value field. Thus, the payload that can be stored in a Value field is bound to \( \leq 253 \) bytes.

### 2.2.1 Attacking RADIUS

As already identified in the original RFC on RADIUS [RRSW97], the protocol is vulnerable to a dictionary attack on the RADIUS secret. The vulnerability is of special relevance in multi-hop wireless networks such as \textit{Wireless Mesh Networks (WMNs)}. Opposed to infrastructure WLAN where APs are connected by wire to the network’s backbone, WMNs have a wireless backbone. Therefore, RADIUS traffic
may be eavesdropped on every wireless hop in between the AP and the RADIUS server. In order to evaluate and demonstrate the practical relevance of this attack, we implemented it and measured the performance of our implementation. Our results corroborate the relevance of the dictionary attack on the RADIUS secret when the RADIUS traffic is not additionally secured.

There are two weaknesses in how the RADIUS secret is used to protect RADIUS communication. First, the same RADIUS secret is used to protect the confidentiality of sent Pairwise Master Keys (PMKs) in a RADIUS Access-Accept, as well as for integrity protection, i.e., computing the Response-Authenticator of the RADIUS packets. Second, the Response-Authenticator (Equation 2.1) is computed by replacing the Authenticator Field with the Request-Authenticator, which is the nonce from a RADIUS request, and appending the RADIUS secret, which is an ASCII passphrase to this packet. The Response-Authenticator is then computed by applying MD5. Dictionary attacks are possible by computing the Response-Authenticator by guessing RADIUS secrets and comparing the results to the expected Response-Authenticator. Since different possible passphrases can be tested in parallel, an offline distributed dictionary attack against the RADIUS secret is possible. Because the RADIUS secret is appended to this buffer and the beginning of the buffer containing the RADIUS packet does not change, it is not necessary to recompute MD5 blocks containing this data.

As MD5 operates on blocks of 64 bytes, the MD5 state can be computed once, and only the last block of MD5 containing the RADIUS secret and the necessary padding will have to be computed for each of the guessed passphrases. This construction of the MD5 authenticator allows to improve the speed of a dictionary attack. Using an Intel Core-i7 870 processor, the speedup of running this optimized attack in parallel against the sequential unoptimized algorithm has been evaluated. A single thread attack allowed a speedup of 3.42. Using two threads, a speedup of 6.92 has been achieved, a speedup of 12.34 using four threads, and a speedup of 15.43 using eight threads. The speed using eight threads is equivalent to testing 36,428,055 different RADIUS secrets per second. Therefore, a network allowing unprotected RADIUS traffic is potentially vulnerable to this attack. An attacker only needs to eavesdrop two packets in order to allow the described attack.

2.3 Wireless LAN

The IEEE standard 802.11 [IEE99] specifies the Medium Access Control (MAC) and the Physical Layer (PHY) for wireless Local Area Network (LAN). The standard adapts the MAC and the PHY to the special requirements of WLANs. For example, the
wireless medium is unprotected from other signals in the environment, so that interference with other signals is likely. Also all signals can easily be eavesdropped by an attacker in range. The latter is addressed by amendment 802.11i [IEE04a] which specifies authentication, key management, and the link layer security between client and access point.

The IEEE 802.11 allows for two operating modes, i.e., the ad hoc and the infrastructure mode. In the ad hoc mode two STAs recognize each other and establish a peer-to-peer association to form an Independent Basic Service Set (IBSS). In the infrastructure mode all STAs connect to a central entity, i.e., the AP. It is able to bridge data between all STAs associated to it. An AP and its associated STAs form a so called Basic Service Set (BSS). Extending, a collection of APs form the so called Extended Service Set (ESS). An ESS uses the same Service Set ID (SSID) across all BSSs. Also, a so called Portal, usually co-located with an AP, may be present. It connects a BSS with other networks, e.g., to the Internet.

In order for a STA to connect to a BSS it has to discover available APs. This can either be achieved by active or passive scanning. Active scanning involves the STA sending out sequential probe requests on each channel. The AP replies with a probe response frame which contains the SSID, Basic Service Set Identifier (BSSID), supported rates, and its security capabilities. The SSID represent the human readable name of the BSS, whereas the BSSID typically relates to the AP’s MAC address.

Passively scanning results in the STA listening for beacons sent out by the APs. A beacon is typically sent every 100 ms.

To associate with the AP the STA needs to initiate the Open System authentication [IEE04a]. The first of two messages is the Open System authentication request. The AP replies with an Open System authentication response which may already indicate success. Next, the STA issues an association request which includes the desired SSID and the supported rates. Finally, the AP confirms the supported data rates and the session ID.

**802.11i Security.** In pre-802.11i deployments the STA would now be able to fully access the network. While private networks typically use pre-shared key mechanisms, enterprise network requires the STA to unlock the 802.1X [IEE10] port, mapped to the newly created association. In order to unlock the port, the STA has to perform an EAP authentication (cf. Section 2.1) using EAP Transport Over LAN (EAPOL) which is a protocol that runs between STA and AP. EAP packets are encapsulated by EAPOL in Layer 2 frames. The message flow is shown in Figure 2.4.

First, the AP (authenticator) sends the EAPOL-Request Identity frame to the STA
2.3. Wireless LAN

![Diagram of EAPOL protocol message flow]

**Figure 2.4:** Message flow of the EAPOL protocol.

(peer). The STA replies with an EAPOL-Response frame which contains its identity. Now, the AAA server and the STA execute an EAP method. An EAP method which is key generating must be used to ensure that after its execution the STA and the AAA server are in possession of the MSK. The AAA server then delivers the MSK to the AP. Upon receipt of the AAA success message, the AP sends an EAPOL-Success frame to the STA.

Now, the STA and the AP are in possession of the so called PMK which consists of the first 256 bits of the MSK. It is subsequently used to create a link layer security association between STA and AP by running the 4-way handshake. The purpose of this handshake is to explicitly confirm that both parties are in possession of the PMK. Also, the PMK is used to derive fresh *Pairwise Transient Keys (PTKs)* to protect the communication between STA and AP on the link layer.

Figure 2.5 shows the message flow of the 4-way handshake. First, the AP picks a random nonce *ANonce* and sends it to the STA. This frame is not yet protected by any security mechanism. Now the STA picks a nonce, called *SNonce* and derives the PTK using the *Pseudo Random Function (PRF)* shown in Equation 2.2.

\[
PTK = PRF(PMK, ANonce || SNonce || AP MAC-Addr || STA MAC-Addr)
\]  

(2.2)

First, the PRF is keyed with the PMK. The second parameter is a concatenation of the exchanged nonces and the MAC addresses of the AP and the STA. After generating the PTK, the STA sends its generated SNonce to the AP. This frame is already integrity protected by a *Message Integrity Code (MIC)*. Now, the AP derives
Figure 2.5: Message flow of the IEEE 802.11i 4-Way handshake.

IEEE 802.11 Beacons. The IEEE 802.11 [IEE99] standard allows to define and transport so-called vendor-specific Information Elements (IEs) which are transported inside of the wireless beacons broadcast by the APs.

Parts of the IEEE 802.11 beacon frame body are shown in Table 2.1. It contains the Robust Security Network (RSN) element which is used to provide information about security features of a device. Last the beacon frame specifies the vendor-specific IEs that are used to transport information that is not covered by the standard. The IE format is shown in Table 2.2.

The IE needs to contain the Identification (ID) of the IE and is set to 221, identifying it as a Vendor-Specific IE. A length field contains the length of the Vendor-/Specific content including the Organizationally Unique Identifier (OUI). The OUI is defined by the vendor to specify the content. A single frame can contain multiple Vendor-Specific IEs which must have a different OUI to determine the subsequent content.

IEEE 802.11u. The IEEE 802.11u [IEE04b] has been amended to the IEEE 802.11 standard in 2012 and intends to improve interworking with external networks, such
as the mobile telecommunication networks. However, it also introduces interesting features for network discovery and AP selection. These new features are to be used in the pre-association phase, i.e., the STA is currently not connected, but scanning for available APs in its surrounding. The APs are supposed to be advertising their network type (private network, free public network, paid public network), roaming consortium (e.g. eduroam [WF05] and so-called venue information. They also provide a Generic Advertisement Service (GAS), which should indicate the offered services using Layer 2 communication between the non-associated STA and the AP. Additionally, IEEE 802.11u introduced the Access Network Query Protocol (ANQP). This protocol can be used to allow the STA to discover additional information about the network in its preassociation phase. They can for instance include:

- the name of the operator,
- available roaming partners at this AP,
- IP address (here the AP), including type and availability

and any other unspecified, but useful meta data which might be helpful to the STA. The above data is to be distributed using the IEEE 802.11 wireless beacons and vendor specific elements.

In the context of our handover protocols (cf. Chapter 7), we propose to use the IEEE 802.11u amendment to distribute this kind of useful meta information to the
Mesh Clients (MCs). Additionally, the concept of FSASD Authenticators (FAUs) (cf. Chapter 5) leverages this beacon functionality.

### 2.4 Internet Protocol Security

The Internet Protocol Security (IPsec) suite has been originally proposed in 1995 in RFC 1825 [Atk95]. Since then, many subsequent RFCs have followed updating the standard [JLN11].

IPsec allows confidential, integrity protected, and authentic IP traffic between two endpoints. This can either be network-to-network or host-to-host. For instance, it is used to securely connect LANs at different branches of enterprises, secure remote access from Internet to Intranet (Virtual Private Network (VPN)), secure connections between individuals host, or routers.

Key security services provided by IPsec are:

- IP-packet-level origin authentication and integrity
- Protection against replay attacks
- IP-packet-level encryption
- Limited traffic flow confidentiality

For this thesis the most relevant parts of IPsec are:

- **Authentication Header (AH)**: authenticity and integrity of IP packets
- **Encapsulating Security Payload (ESP)**: confidentiality, integrity, and authenticity of IP packets
- **Security Association (SA)**: collection of key and cryptographic algorithms to be used with AH and ESP, SA establishment, e.g., using Internet Key Exchange Protocol (IKE)

An IPsec SA defines how Internet Protocol (IP) packets are to be secured between two endpoints. It consists of the destination IP, the Security Policy Identifier (SPI),
2.4. Internet Protocol Security

IPsec can be used in transport and tunnel mode. Transport mode only protects the IP packet payload. For example, AH in tunnel mode will encapsulate a packet using the authentication header and IP header. The packet is now authenticated and integrity protected. ESP in tunnel mode encapsulates the whole original packet in a new IP packet. Both, header and payload are secured. In any case, the unencrypted new tunnel headers are used to route packets from one endpoint to the other. The differences in encapsulation and which parts of the packet are secured in which manner are shown in Figure 2.6 and Figure 2.7.

AH can be used in both modes and is used to authenticate source and destination host of the communication. It protects the integrity of the IP packet payload and its headers by using a keyed hash function. This Message Authentication Code (MAC) is included in the Authentication Header of the new packet.

ESP can also be used in both modes. It achieves source authentication, integrity and confidentiality, by adding a header, and a trailer including a MAC to the packet. The original IP payload is fully encrypted. In tunnel mode, also the IP header of the original packet is encrypted, thus, it provides some IP address privacy for traffic that is tunneled through an IPsec gateway.
Entity Authentication & Key Agreement. Since all mechanisms described above need cryptographic material, IPsec provides several key agreement and management protocols. For instance, one can install keys to be used with an SAs, i.e., inject a preshared key. In addition, there are protocols such as IKE, Internet Key Exchange Protocol v2 (IKEv2) [Kau10], Internet Security Association and Key Management Protocol (ISAKMP) [MSST98] to allow automated key agreement and management for IPsec. The described protocols, message formats, and payloads for authenticated key exchange are used to establish full-fledged IPsec SAs.

An often voiced annoyance of IPsec is its complicated usage and key establishment which, additionally, is spread over multiple RFCs. This makes it particularly hard to use and implement, thus, likely inducing misconfiguration and defective implementations. For this purpose IKEv2 [Kau10] has been condensed into one single RFC. It has also reduced the overall protocol complexity by reducing the number of different phase one key exchanges. A directly usable SA is generated during the initial phase one exchange. Phase two exchanges are reduced to two messages with less complex parameters.

In this thesis, we use IPsec by injecting a shared secret key established during authentication or a subsequent authentication and key agreement protocol.

2.5 The pwrcall RPC Framework

In this section we shortly discuss pwrcall, a framework for "secure and lightweight distributed function calls" [Sch11]. It is built onto the foundations of [MYS03] and [Spi07] using the simplicity of capabilities.

Capabilities Model

Capabilities have first been described in [DVH66] resembling an unforgeable token which is used to authorize access to an object. Since then the concept has made its way to operating systems and programming languages [Har85, SSF99, Mil06]. The object capability model of [Mil06] does not differentiate between subjects and objects, but rather defines so-called non-primitive objects as a (code, state)-tuple. This state represents a set of references to certain other objects. Objects can send messages to each other and exhibit a certain behavior which is defined by their programming code. The messages between objects can carry references, thus, influence their computations. The design of [FN79] merges this notion with the functionality that objects know which type of access to it is allowed. This means that objects themselves can expose certain functions as capabilities to other objects to access. In
order to access a certain resource it is required to possess the respective capability referencing the object.

Design Overview

In the following we describe the design of the pwrcall framework. The key features of the pwrcall framework are:

- using object-capability model,
- capability generation is based on symmetric encryption,
- revocable capabilities,
- function calls to third-party objects via delegation.

Functionalities of pwrcall objects are exported as methods of the respective objects. To access them remotely, other objects need to possess a capability to do so. Pwrcall objects are registered within the framework, making it possible that a capability can be passed on to other entities. They can be used to obtain connectivity by initial condition when passed out as so-called off-line URLs (below):

```
pwrcall://e7bcae69e9c79aad2f4b8fe1f14bcd52beb4faae0@137.226.161.211:10000,192.168.1.4:10000
/18ed767d925d54a19d0b4a4c4633d0028751eab43
```

Pwrcall specifies a minimum set of parameters needed to enable a remote function call. In order to enable multiplexing the pwrcall messages use IDs. Additionally, a function call must contain the capability designating the remote object, a function name specifying the object’s method call and function call parameters. The response carries an error description, or the function’s return value.

Pwrcall Capabilities

Pwrcall introduces a new way to define capabilities using symmetric encryption techniques, i.e., AES with an 128-bit key size.

\[
\text{Capability} = \text{AES-128} (\text{nodesecret}, \text{nonce}|\text{designator}|\text{options}) \tag{2.3}
\]

The capability itself is created as a cipher text of the parameters nonce, designator, and options (cf. Equation 2.3). A designator is a reference use to look up the target object in the pwrcall framework. As the encryption of the parameters uses a per-node secret key the designator itself does not need to be unforgeable. A nonce enables to set a lifetime of the capability, effectively bound to the node. The options structure allows for automatic revocation after a period of time.
Communication Design

Pwrcall exposes a Node as its central component being responsible to host objects intended to be accessible from remote systems. It handles serialization, data structures, method-calls, and provides connectivity to other nodes and objects.

It is important to note that pwrcall allows method calls in a bi-directional way. There may not only be one server and several clients, but also multiple nodes communicating with each other’s functions, using their respective objects.

Security Design of pwrcall

Pwrcall allows authenticating nodes and providing access control to objects in a secure way. The TLS protocol is used to provide confidentiality and integrity of communication between nodes and also for authenticating peers. Each peer must present a certificate (in a generic sense) to its communication partner. It is possible to use a Public Key Infrastructure (PKI) to validate certificates, or enable mutual authentication by using known public key fingerprints. Although pwrcall is flexible in this regard, its default usage pattern is the Off-line-URL modus. As access control to objects is secured using object capabilities, access to functions of objects is only possible by using a legitimate reference.

Pwrcall’s security architecture protects against common attacks on network protocols. Using TLS can prevent man-in-the-middle attacks when at least one communication party is correctly authenticated. In pwrcall this is enabled by verification of peer certificates or the use of Off-line URLs verifying the node fingerprint.

In order to successfully forge function call requests, an attacker would need to break the symmetric encryption used to generate the capability. Choosing a strong scheme and large random numbers makes brute-force attacks infeasible as well. The nonces of capabilities hinder replay attacks aiming to use previously valid capabilities.

2.6 Summary

This chapter introduced the relevant technical means, protocols, and concepts which this thesis is based on. After first presenting the most important part, the EAP (cf. Section 2.1), we describe the backend technology, RADIUS (cf. Section 2.2), necessary to fully use the protocol framework. We also discussed the most significant security issues of RADIUS, i.e., the insufficiently strong cryptographic functions and the insecure use of RADIUS secrets and message parameters. These security challenges also motivate the subsequent attack on RADIUS traffic that can be eaves-
dropped if the attacker is located in the correct network segment. Next, we showed how both, EAP and RADIUS, are used in WLAN (cf. Section 2.3) today for entity authentication. This also entailed the discussion of derived keys which are subsequently used to secure the one-hop link layer connection between STA and AP. In Section 2.5 we shortly discussed pwrcall, a secure RPC framework based on the object-capability model. Last, we shortly introduced IPsec (cf. Section 2.4) as a general multifaceted concept of securing host-to-host and network-to-network traffic on the network layer.

All these techniques will play an important role on the comprehensive security architecture for WMNs proposed in this thesis.

Next, Chapter 3 presents the network model, its entities and the application scenarios this thesis assumes as well as the derived requirements a comprehensive security architecture has to fulfill.
Security Challenges of Wireless Mesh Networks

Previous publications identified the most serious security challenges arising from WMNs to be secure multi-hop routing, detection of corrupted nodes, and fairness with respect to the distribution of network resources. While there is consensus on the importance of these challenges, prior work has not yet led to generally accepted security requirements. In particular, existing proposals for security architectures for WMNs tend not to address them as a whole, but rather concentrate on one or two specific characteristics and design mechanisms that meet the security requirements for these specific characteristics (e.g. secure routing or client authentication).

In this chapter we introduce the necessary fundamentals which this thesis is based on. We introduce IEEE 802.11-based [IEE99] WMNs in a general sense, different scenarios and operator models, as well as the involved entities and node types. Additionally, we define the communication patterns of such networks and characteristics which we consider important in terms of security. Next, we define the attacker model which is used in this thesis. We then use this framework of requirements and characteristics to evaluate state of the art proposals ranging from mesh standardization effort of the IEEE [IEE11] to results from academia.

Parts of this Chapter have been published in [EM10].

Outline: Section 3.1 summarizes previous work on security challenges and requirements for WMNs. Section 3.4 highlights the relationship between the characteristics and the security requirements by discussing the challenges arising from each of the characteristics. In Section 3.6 we evaluate previous proposals with respect to the scenarios and characteristics they support, as well as the security requirements they meet and conclude our work with Section 3.7.

3.1 Security Architectures and Mechanisms for WMNs

With respect to exploring the unique characteristics, imposing security threats and requirements some researchers provide an overview of the current state of the art
of WMN security.

The security challenges for WMNs were first investigated by Salem and Hubaux [BSH06]. They identified three major security challenges, namely:

- the detection of corrupted nodes,
- secure multi-hop routing, and
- fairness with respect to the distribution of network resources.

These challenges were confirmed by Siddiqui et al. [SH07], Khan et al. [KMLS08], and Glass et al. [GPM08]. The results of Siddiqui et al. [SH07] exceed Salem and Hubaux [BSH06] in the sense that the authors also provide a notion of threats WMNs face, e.g., Denial of Service (DoS) attacks. In addition, they discuss proposals to act on these threats on an abstract level. Kahn et al. shift their focus to passive security threats for WMNs such as determining the gateways of a network [KMLS08]. Glass et al. focus on threats related to the MAC layer and the routing protocol [GPM08]. With respect to application scenarios only Salem and Hubaux [BSH06] point out that WMNs may also be operated by multiple operators. They identify mutual authentication of devices from different administrative domains as well as other charging policies of operators to be additional challenges.

In contrast to previous work we generally specify the core scenarios and characteristics of real world applications of WMNs along their relevant communication patterns. For the specification of these scenarios we take real world applications that use hardware, e.g., provided by companies such as Cisco-Meraki 1, MeshDynamics 2, and Tropos 3 into account. In addition, we consider the work of [MLR+02, SHJB07], and [PP08], which prove valuable for identifying the characteristics of WMNs.

While the challenges pointed out by [BSH06, SH07, KMLS08], and [GPM08] are broad consent in the literature, a more detailed, generally accepted list of high-level security requirements for WMNs is still missing. As a consequence, it is hard to evaluate the strengths, weaknesses, and open issues of existing proposals. We specify a comprehensive set of high-level security requirements and discuss how the potential characteristics of WMNs influence the design of security mechanisms to meet these requirements.

The identified security requirements allow us to analyze previously suggested security architectures for WMNs. While a large variety of such proposals exists (e.g., [CLMC06, MPC08, ZF06, RL08, IYHH08, KJ08, BD09, HA10, WMLW11, SZZF11]),
3.2 Wireless Mesh Networks

One of the most important technologies to provide connectivity, e.g., to the Internet, is IEEE 802.11 [IEE99] which this thesis assumes to be the link layer technology. However, all our proposed security mechanisms can easily be used in wireless networks with different PHY/MAC layer, e.g., WiMAX [IEE12]. IEEE 802.11 WLAN is used in the public sector as well as in enterprises\(^4\). It typically involves an operator providing infrastructure support, namely access points connected by wire to a backbone. Providing a wired infrastructure is a costly endeavor, needs careful planning, and results in a static structure. WMNs enable wireless communication between infrastructure components of the network. WMNs thus make an abundance of wires obsolete, leading to a flexible and potentially dynamic network infrastructure. Figure 3.1 illustrates a typical WMN infrastructure.

3.2.1 Node Types

We consider that the WMN consists of the following node types:

---

\(^4\)http://www.tropos.com/
Mesh Clients (MCs) are the users of the WMN. They are typically end-user devices such as Laptops or Smartphones which connect to the network infrastructure such as Mesh Routers (MRs). In some cases, MCs may also be able to route network traffic.

Mesh Routers (MRs) are the most used infrastructure nodes of WMNs. MRs connect to MRs, Mesh Gateways (MGs), and other possibly routing MCs. Within the network all traffic is routed by MRs.

Mesh Gateways (MGs) are part of the infrastructure which provide access to other networks, typically the Internet.

Authentication, Authorization and Accountings (AAAs) servers are responsible for authenticating the nodes participating in the WMNs. Typically this is based on a AAA protocol such as RADIUS [RRSW97].

Network Access Servers (NASs) are a role any of the above nodes (except the AAA server) can assume. It involves acting as authenticator services to other nodes in conjunction with the AAA server.

### 3.2.2 Communication Patterns

WMNs have to support the following communication patterns between the nodes in the network:

1. MC ↔ MC communication between two clients (located in the same WMN).
2. MC ↔ MR communication refers to that of clients and the associated access point.
3. MC ↔ MG communication refers to multi-hop traffic destined to leave the WMN through the gateway. This may also include management traffic, e.g., when communicating with a AAA server located outside of the WMN.
4. MR ↔ MR communication refers to all traffic between MRs.
5. MR ↔ MG communication can be considered as special cases of the former. It may include management traffic, but also forwarded user traffic.

While the aforementioned communication patterns assume unicast traffic, broadcast traffic also needs to be considered. Such traffic most likely emanates from central network components, e.g., MGs or administrative entities such as key servers. In context of IEEE 802.11s [IEEE11] amendment, Root Announcements (RAANs) are propagated through the network in order to provide link metrics to its routing protocol.
Considering this example, there may be local as well as global broadcast traffic. For instance, if the routing protocol relies on a global network map, it may need to be broadcast through the network. Beacons of MRs can in contrast be considered as local broadcast. Most of the assumable broadcast traffic will be related to the routing protocol, but also other possibilities such as synchronization may exist.

3.3 Attacker Model

For this thesis we consider two types of attackers: external and internal attackers. Before detailing both type of attackers, we define their common capabilities which implicitly represent active and passive capabilities:

- Manipulation or malicious jamming of the physical medium;
- (Selectively) overhearing link layer traffic;
- Inability to break sufficiently strong cryptographic ciphers and hash functions [DY83].

External attackers do not possess the necessary credentials to obtain legitimate network access. Their goal is either to gain network access to mount subsequent attacks (as an internal attacker), or to maliciously disrupt parts of the network. Thus, they mount attacks from outside of the network, either on the physical layer, or the network’s link layer. They are able to eavesdrop, inject and manipulate network traffic on the link layer between legitimate network nodes. Also, the external attacker can initiate a conversation on the link layer with any node in range of its current position.

Internal attackers are legitimate, authenticated and authorized devices (MRs, MGs and MCs) of the network. Additionally, external attackers may have succeeded in compromising a legitimate node, thus becoming an internal attacker. The goal of an internal attacker is to learn about the communication inside of the network, that of a specific node, or that of communication partners. For instance, an attacker using a compromised node is able to eavesdrop, manipulate and disrupt any communication flowing through the node. More specifically, an internal attacker’s goal involves compromising other nodes, e.g., by eavesdropping key transport.

Colluding Attackers. Collusion may happen between external, as well as internal attackers. While the impact of cooperating external attackers is most likely restricted to disrupting specific regions of the network, colluding internal attacker can pose a significant threat. By increasing the amount of compromised nodes, the likelihood to
successfully mount eavesdropping attacks which compromise confidentiality, rises. Also, impersonation attacks can be more effective if, e.g., some nodes compromise key material, while other nodes use this material to impersonate nodes.

3.4 Scenarios, Characteristics & Challenges

WMNs are mainly studied in the context of two distinct manifestations:

- **single-operator** scenarios in which a single operator provides and maintains the infrastructure, and
- **multi-operator** scenarios in which multiple operators provide and maintain the infrastructure.

The latter can further be characterized by the fact that they support roaming of MCs between the networks operated by different providers and scenarios that may additionally support infrastructure sharing. Infrastructure sharing allows different operators (also private individuals) to contribute to a single network that consists of their collective devices. In both scenarios, communication between all types of nodes (MCs, MRs, and MGs) has to be supported. Additionally, in single-operator as well as multi-operator scenarios WMNs may support routing MCs, mobility of the infrastructure nodes, and heterogeneity in the technologies applied on the different wireless communication links. In the rest of this section we discuss these scenarios and characteristics in more detail.

3.4.1 Single-Operator Wireless Mesh Networks

In single-operator WMNs all infrastructure nodes are controlled and maintained by a single operator. Typical applications include intelligent transportation systems, public safety support, Internet access, smart metering, and building automation.

The operator is responsible for the deployment of the network infrastructure, but not necessarily the MC hardware. MR and MG hardware provided by the operator is typically homogeneous. The operator is able to influence the topology of the network except for the mobile or stationary MCs. With respect to network access, MCs are typically required to perform initial registration with the operator (e.g., obtain access credentials, disclose personal details and payment information, etc.). This allows the network operator to restrict the access to the network. In order to provide such access control the network operator can employ a AAA system, e.g., based on protocols such as RADIUS [RRSW97] or Diameter [PJE+03]. If the
network has an up-link to the Internet, the AAA server could also be located outside of the WMN.

In single-operator scenarios the traffic within the WMN must secure all communication patterns mentioned in Section 3.2.2. Even though the devices are operated by a single operator, they cannot be fully trusted as they may be compromised. This is due also due to the fact that some devices might be placed in easily accessible areas.

### 3.4.2 Multi-Operator Wireless Mesh Networks

In multi-operator WMNs, several operators provide and maintain infrastructure components. In the simplest case, each operator maintains a separate network but the clients registered with any of the operators may roam to WMNs provided and maintained by other operators. Possible applications of WMNs interoperation in such a way include the previously introduced single-operator scenarios, e.g., Internet access or building automation. Here, access control needs to ensure that MCs of interworking operators are able to access a network without being registered to the operator of the network they currently want to access.

In more evolved multi-operator scenarios, operators may want to share some of their infrastructure components with each other. For example in building automation different companies may be responsible for their respective MRs while sharing the Internet up-link through the same MG. Other examples are temporary venues such as concerts and trade shows where the organizer of the event may operate several MGs and all exhibitors run their own MRs for their own clients while sharing the MGs. In addition, disaster recovery is an important area which is mostly known to be an application area of ad-hoc and sensor networks [MLR+02]. However, WMNs can be used to converge multiple networks and provide improved resilience, bandwidth and coverage.

In contrast to a single operator the different operators will typically have heterogeneous hardware components which they contribute to the network. Thus, there needs to be a way to coordinate the initial setup of the network. With respect to the backbone of the WMN, each operator may maintain its own, or offer it as a service to other operators. If there is no prior trust relationship between the operators, mutual authentication across multiple domains is challenging. In addition, operators can have different policies for providing network access to visiting clients [PP08].

In the simplest multi-operator scenario, where only MCs are able to roam between operators, the main challenge for meeting the security requirements is enabling access to the WMN of an operator based on the registration with another operator. However, this is a challenge well-known from other wireless technolo-
gies such as WLANs and cellular networks. In addition, protecting the privacy of roaming MCs is a challenge in a multi-operator environment. This is particularly challenging if identity privacy of the MC in the Foreign Network (FN) is desired. Enabling non-repudiation of service usage is also more challenging in multi-operator scenarios.

When considering more evolved multi-operator scenarios where the operators share part of their infrastructure, access control for newly connecting nodes becomes challenging. Particularly, newly joining MRs, and MGs will now have to establish keys with infrastructure nodes operated by different operators. Also, fairness becomes more of an issue as in addition to per-MC fairness, per-operator fairness (potentially proportional to the infrastructure contributed by the operator) in the distribution of bandwidth is required.

Chapter 6 proposes a framework specifically addressing the challenges of multi-operator WMNs. It provides an extension of the comprehensive security architecture for WMNs proposed in Chapter 5.

### 3.4.3 Heterogeneity of the Wireless Links

WMNs do not necessarily have to use a specific or the same technology between all network entities. MCs could for example use wireless technologies such as IEEE 802.11 [IEE99], 802.15.4 [IEE07], or 802.16 [IEE12] depending on the properties of the WMN’s MRs and different wireless technologies could be used between MCs and MRs and between MRs.

Heterogeneity of the wireless links can make the protection of the communication between non-neighboring nodes more challenging as it will require the use of integrity and/or encryption on a higher protocol layer than the data link layer. In addition, as not all wireless technologies typically support the same cryptographic algorithms, heterogeneity may lead to links that provide a different level of protection. This again leads to the requirement that MC-generated traffic destined to other MCs or to external nodes should not only be protected hop-by-hop.

The security architecture and mechanisms proposed in this thesis is PHY and MAC layer agnostic. As such, the technologies to secure the single hop link communication between, e.g., MC and MR may differ, but the key management does not.

### 3.4.4 Routing Clients

In single- as well as multi-operator scenarios MCs can act as legacy clients only or may additionally take the role of a MR as well. As such, other MCs or parts of
the infrastructure may associate with the MC which then relays their traffic. The operator’s gain to allow MCs to act as MRs is that of increasing the coverage of the network without deploying more hardware himself.

Apart from the challenge to provide incentives for MCs routing traffic for other MCs, the main security challenge arising from routing MCs is that from another MC’s perspective, they cannot be trusted. Unfair behavior can be induced by routing clients. This is due to the fact that the routing client could try to mask his traffic as originating from other clients communicating via his link. However, this challenge is already captured in our security requirements by the fact that we do not require an MC to trust its MR with respect to the adequate protection of the traffic it generates. As we assume MRs to be untrusted, a routing MC does not impose new requirements.

This thesis does not specifically focus on routing clients or secure routing for WMNs. However, the comprehensive security architecture for WMNs proposed in Chapter 5 automatically supports routing MCs by treating them equivalently to untrusted MRs. Regulatory and accounting issues introduced by such a functionality are not in the scope of this thesis.

### 3.4.5 Mobility

The infrastructure nodes (i.e., MRs, MGs) can be static, i.e., deployed and thereby added to the network in one location and then operated for a longer period. However, in some applications enabling mobility of the infrastructure nodes may be of interest as well. In particular, this includes industrial application scenarios, e.g., underground mining [ATOF04], where MRs and MCs may move together or even MCs are static while MRs move, or resilient networks for first responder units [PSBM11].

Enabling infrastructure mobility imposes several challenges for the security mechanisms. The access control for MRs, and MGs has to be flexible (e.g., not involve interaction with the operator) and more efficient as they need to be executed more often than only during the initial network attachment. Also, any mechanism used to ensure fairness (e.g., bandwidth and latency) has to cope with infrastructure mobility.

In this thesis we consider both mobility of end-users and that of parts of the networks’ infrastructure. Chapter 7 specifically deals with secure, efficient, and practical proactive handover protocols that can be used by any node of the network, regardless of it being a client or part of the network infrastructure.
Table 3.1: An overview on the security requirements for the different communication patterns in WMNs.

### 3.5 General Security Requirements

In this section we describe the general security requirements for WMNs with respect to confidentiality, integrity protection, replay protection, access control, privacy, availability, fairness, and non-repudiation. All of these general requirements are relevant in WMNs independently of the characteristics discussed in Section 3.4. However, as we will show in Section 3.4 the difficulty of designing mechanisms that meet the requirements depends on the characteristics of the WMN in question.

### Protecting the Communication

The communication over wireless links in WMNs has to be protected against eavesdroppers as well as against replay and manipulation. Due to the multi-hop nature of WMNs and the fact that infrastructure nodes may be easy to compromise, a simple hop-by-hop protection is not sufficient. In the following we provide a detailed analysis of the communication patterns in WMNs and argue for each of them which type of protection should be required or optionally provided for them. Table 3.1 summarizes the results of our analysis with respect to confidentiality, integrity, and replay protection for all communication patterns.
3.5. General Security Requirements

Definition 1 (Confidentiality) "Confidentiality is the concealment of information or resources." [Bis04]

Confidentiality is considered optional between two MCs to prevent intermediate MRs, MCs and outsiders from eavesdropping on the communication. While encryption between MC and MR would prevent eavesdropping on the initial wireless connection, it does not safeguard against eavesdropping by other MRs, MCs and outsiders located on the multi-hop path segment after the initial hop. Communication of any node with the AAA server must be confidential as network access credentials or keying material might be exchanged. We require MC ↔ MG communication to be confidential. This counters eavesdropping threats originating from intermediate MRs, and other MCs. Additionally MR ↔ MR communication may also be confidential. For example, if user traffic is confidential between MC ↔ MG, information who communicates with whom can still be leaked by the routing protocol. With respect to broadcast traffic, not all the traffic needs to be kept confidential. However, some traffic will require confidentiality of broadcast messages. For instance, broadcasting network maps and link metrics in plaintext can be considered a threat. An attacker could gain insight of the network topology and the available routes to facilitate a subsequent attack on the network. The attacker could for example try to disable specific MGs.

Definition 2 (Integrity) "Integrity refers to the trustworthiness of data and resources, and it is usually phrased in terms of preventing improper or unauthorized change. Integrity includes data integrity (the content of information) and origin integrity (the source of the data, often called authentication)." [Bis04]

Definition 3 (Replay) (Message) Replay refers to the process of using a previously captured message and replaying it to the receiver.

Integrity- and replay-protection are both important for all the introduced communication patterns. Just as confidentiality, integrity and replay protection are both required between two MCs, MC ↔ MR, MC ↔ MG, and any node and the AAA server to prevent unnoticed manipulation. However, integrity and replay protection are also required for MR ↔ MR as well as MR ↔ MG communication. Note that one can argue that assuming confidentiality of MC ↔ MG communication, integrity without additional confidentiality is sufficient on the first hop between MC ↔ MR. The MR can simply check whether the MC’s traffic is allowed to pass. Within the WMN integrity can be attained between MRs in a hop-by-hop fashion and the MRs checking the traffic for its legitimacy. Integrity and replay protection are also of importance when considering broadcast traffic in order to prevent manipulation or the
distribution of outdated information. If for example outdated routing information would be distributed, the MRs could not provide valid routes.

**Definition 4 (Access Control)** “When a system mechanism controls access to an object and an individual user cannot alter that access, the control is mandatory access control (MAC), occasionally called a rule-based access control.” [Bis04]

Access Control entails authentication and authorization of network entities. It is required to control which entities are allowed to access the network. Entity authentication should be combined with key establishment to bootstrap integrity and encryption mechanisms and thereby also ensure ongoing access control. Authentication is equally important for users and operators, since users need to ensure that the network is the one it claims to be, as well as vice versa. Access control is required for all network entities, such as MCs, as well as newly joining MRs, and MGs.

**Definition 5 (Privacy)** “The right to privacy is our right to keep a domain around us, which includes all those things that are part of us, such as our body, home, property, thoughts, feelings, secrets and identity. The right to privacy gives us the ability to choose which parts in this domain can be accessed by others, and to control the extent, manner and timing of the use of those parts we choose to disclose.” [Onn05]

Privacy is often falsely thought of as a goal that is automatically achieved alongside confidentiality. However, privacy issues can for example arise when authenticating an MC to an MR. Although the communication can be kept confidential between MR and MC, the MR could still learn identity attributes of the MC. In context of mobility and repeated authentication, tracking also becomes an issue that cannot solely be solved by keeping the communication between MC and MR confidential.

### 3.6 Security Architectures for WMNs

In this section we evaluate prominent security architectures proposed for WMNs with respect to their characteristics, the scenarios they support, and which security requirements they meet. This includes a detailed discussion of the mechanisms they apply to meet the requirements. We concentrate on four major proposals which try to be of a comprehensive nature, the IEEE 802.11s [IEE11] standard and three more research driven proposals, such as the Attack Resilient Security Architecture for Multi-hop WMNs (ARSA) [ZF06], the MobiSEC [MPC08], and a PANA-based5 architecture [CLMC06]. These four proposals are evaluated in detail.

---

5 Protocol for Carrying Authentication and Network Access
3.6. Security Architectures for WMNs

and compared at the end of this section. A summary of our findings is provided in Table 3.2. In addition to these four more comprehensive proposals for WMN security architectures, we briefly summarize proposals that concentrate on solving one particular security challenge such as key establishment.

3.6.1 IEEE 802.11s - Mesh Networking

When discussing proposed security architectures for WMNs, it is of course important to consider upcoming standards such as the IEEE 802.11s. After being successfully passed, network equipment vendors will implement it and roll out their hardware with wireless mesh networking support at the time of writing. It supports access control for all types of nodes (MCs, MRs, MGs) based on two protocols: the Simultaneous Authentication of Equals (SAE) protocol and the Abbreviated Handshake protocol. The former is a password-based authentication protocol that allows two arbitrary types of nodes to simultaneously authenticate each other and establish a PMK. SAE thus assumes a pre-shared secret, namely a password, to be known to all legitimate network nodes. The Abbreviated Handshake is used for authentication and key agreement between peers that already share a PMK, i.e., a pair of peers that have already successfully run SAE before. The Abbreviated Handshake protocol requires fewer messages to be exchanged between the nodes than the SAE protocol, which explains its name. Keying material generated during the Abbreviated Handshake protocol is subsequently used to encrypt, integrity protect and replay protect the communication between the nodes. In the following we detail both authentication and key agreement protocols and discuss their shortcoming.

Simultaneous Authentication of Equals

The computations used by SAE are either based on Elliptic Curve Cryptography (ECC) \cite{HMV03} or prime modulus finite cyclic groups. In the following we use the notation of ECC-based SAE in which \( P(x, y) \) represents a point on a publicly known elliptic curve of the form

\[
y^2 = x^3 + ax + b.
\]

By \( inv \) we refer to the additive inverse of a point on the elliptic curve. SAE uses four messages to authenticate two peers in a simultaneous fashion. The message flow of SAE between parties A and B is depicted in Figure 3.2. In the first step the initiating peer generates a Password Element (PWE) which represents a point on an elliptic curve. The PWE is combined with a hash \( m \) containing a combination of MAC addresses of the respective two peers by scalar multiplication to
\[ N = \text{PWE} \times m. \]  

(3.2)

The initiating peer \( A \) constructs a commit scalar \( cs_A \) (cf. Equation 3.3) and a commit element \( ce_A \) (cf. Equation 3.4).

\[
\begin{align*}
   cs_A &= (\text{rand}_A + \text{mask}_A) \mod \tau \\
   ce_A &= \text{inv}(\text{mask}_A \times N)
\end{align*}
\]

Here, \( \text{rand}_A \) refers to a random number which is used to create the commit scalar \( cs_A \). \( \text{mask}_A \) is a temporary secret value used to blind the transferral of the random number. Upon reception of a peer’s commit, both peers are able to compute the same secret \( k \) using a pre-defined key derivation function \( F \). \( k \) is derived by each peer based on the other peer’s commit message, its own random number, and \( N \) such that \( A \) computes Equation 3.5 and \( B \) computes Equation 3.5.

\[
\begin{align*}
   k &= F((\text{rand}_A \times (cs_B \times N + ce_B)) \\
   k &= F((\text{rand}_B \times (cs_A \times N + ce_A))
\end{align*}
\]

The computation effectively represents a password-authenticated ECC Diffie-Hellman key exchange [DH06]. Both peers will build a confirmation message, namely a hash of the secret \( k \), a replay-protection counter (\( \text{ctr} \)) and the previously exchanged \( cs \) and \( ce \) values. If the received confirmation message equals the expected result, authentication is considered successful. After the authentication was successful, both peers will generate a pairwise master key as:

\[
\begin{align*}
   \text{PMK} &= H(k|\text{ctr}||(cs_A + cs_B) \mod \tau||F(ce_A + ce_B))
\end{align*}
\]

If a PMK has been successfully established, it can be used during the Abbreviated Handshake. The PMK is used to construct a key hierarchy in which a 128-bit Abbreviated Handshake Key Confirmation Key (AKCK), a 256-bit Abbreviated Handshake Key Encryption Key (AKEK), and a 128-bit Mesh Temporal Key (MTK) are computed. The key derivation is based on an exchangeable PRF which produces 256 bits of key material. The keys AKCK and AKEK are static in the sense that they can be used to provide origin authenticity and data confidentiality in multiple runs of the Abbreviated Handshake and Group Key Handshake. AKEK is used to encrypt the Group Transient Key (GTK) during the Abbreviated Handshake. The MTK is used to protect the communication between two peers and is derived in a more dynamic manner by using freshly generated random numbers of both peers as input to the
key derivation function. The PMK, AKCK and AKEK’s lifetime is limited by the password’s lifetime, whereas the MTK should be regenerated each time the session between two nodes times out.

**Abbreviated Handshake**

The goal of the protocol is to generate a fresh MTK between two peers that already share a PMK due to having been connected to each other before. The MTK key generation using the PMK is randomized by using two fresh random numbers selected by the peers. Since the peers share a PMK and therefore AKCK and also AKEK, the exchange of the nonces can be integrity protected. The protocol consists of two messages, i.e., a **Peering Open Frame** which also contains the random number and a **Peering Confirm Frame** containing the nonce of the respective other peer.

**Analysis**

IEEE 802.11s provides access control to the WMN for all types of nodes which is combined with mutual authentication and key agreement. The keying material generated can be used to meet all security requirements with respect to confidentiality, integrity, and replay protection in WMNs. However, access control is based on a
single password shared between all legitimate nodes. This leads to a number of shortcomings:

- The compromise of one single node in the WMN enables other unauthorized nodes to join the network. This is due to the fact that the lifetime of the PMK is bound to the lifetime of the password. And the PMK itself is used as the root of the key hierarchy.

- Since the only identity attribute used in the protocols is the MAC address of a peer, impersonation of arbitrary peers by others is possible. For instance, legitimate nodes can impersonate other legitimate nodes.

- As routers are also considered to be peers, just as clients, an attacker in possession of the password can impersonate the network to a client.

- Finally, excluding a specific client or router from the network is not possible as the operator would have to change the network access control password and restart the network.

In addition, due to the use of passwords as the main access credentials, the standard neither supports roaming of MCs between multiple operators, nor sharing of network nodes between different operators. Mobility of infrastructure nodes could be supported with the help of SAE and the Abbreviated Handshake protocol.

3.6.2 Attack Resilient Security Architecture for WMNs

Zhang et al. proposed an Attack Resilient Security Architecture for Multi-hop Wireless Networks (ARSA) that aims at providing secure roaming in multi-domain WMNs based on so-called passes that are linked to trusted brokers [ZF06]. They employ Identity-based Cryptography (IBC) [BF01] in order to circumvent broadcasting lengthy X.509 [AF99] certificates. IBC also enables self-authenticating public keys since they can be reproduced by anyone knowing the identity, e.g., based on the Network Access Identifier (NAI) [ABE05], of the entity and the domain parameters. Brokers issue signed passes to MCs. If a MC accesses a WMN, the operator must have an agreement with the broker in order to support the MC, i.e., for billing. Once the MC provides the pass issued by his broker, the included public key is used to encrypt a temporary network access pass issued by the respective operator. The client checks network legitimacy by verifying the signature on the operator’s domain parameters. Domain parameters are much like certificates in context of IBC, since they provide means to gather the cryptographic parameters necessary to perform validity checks. MC to MC authentication (in the same operator’s network)
is based on temporary passes issued by the operator. The possession of such a pass is assumed to be sufficient to trust the MC as both MCs have been previously authenticated by the operator.

Analysis

ARSA explicitly supports access control of roaming and non-roaming MCs to WMNs. However, ARSA does not address the protection of MC ↔ MR, MC ↔ MG, MR ↔ MG, and MR ↔ MR communication. In addition, ARSA does not address access control for MRs and MGs, and consequently does also not support infrastructure mobility. Also, routing MCs are not supported by ARSA.

An additional specific shortcoming of ARSA lies in the fact that in order to function properly the MRs need to be pre-loaded with router passes (to prove affiliation to the operator), as well as passes of trusted brokers. If broker passes are missing, they are supposed to be provided by the MCs. The authors, however, fail to mention how to decide whether to trust the broker or not. Domain parameters of operators are signed by Trusted Third Parties (TTPs). This does, however, involve MCs having the necessary parameters to verify the signature.

The most serious problem stems from the broker passes themselves. If a client loose his pass, he is supposed to report this to the broker. But since it can be considered hard to notice that a pass has been stolen, reporting will most likely not take place. Attackers may use this fact to impersonate a MC since authentication is solely based on this pass. Since the broker passes are fixed for each MC, tracking issues arise because of passes being used each time network access takes place.

3.6.3 MobiSEC

The MobiSEC proposal of Martignon et al. features access control to a WMN for MCs as well as for MRs [MPC08]. The key idea is to use IEEE 802.11i [IEEE04a] for authentication and key agreement between any node (MC or MR) in the role of the supplicant and the MR it attaches to as authenticator. In a second stage authentication, the node can use a protocol based on TLS and a Certification Authority (CA) signed certificate with the AAA server to additionally authenticate as a router and obtain the keying material required for this role in the WMN. Note that the authors suggest that all infrastructure nodes use the same network-wide backbone key to protect their communication in order to facilitate seamless mobility for MRs without re-authentication. The authors also propose two options for a re-keying mechanism for the backbone key: re-authentication with the key server and an option where each infrastructure node obtains a seed value from the key server to
generate the new key locally.

Analysis

While MobiSEC addresses access control to the WMN including authentication and key establishment for MCs as well as MRs, confidentiality and integrity and replay protection are not explicitly addressed in the architecture. In particular, the proposal only supports the protection of MC ↔ MR and MR ↔ MR communication and none of the other communication patterns. Multiple operators are not explicitly addressed such that the support of roaming MCs and infrastructure sharing is unclear. Infrastructure mobility and routing MCs, however, are addressed.

A more detailed analysis of the author’s concepts reveals various security threats. Using a network wide key for the protection of all backbone communication is the most serious one. The compromise of a single MR is sufficient to be able to derive this key. As user traffic seems to be protected in a user-specific way only on the first hop between MC and MR the compromise of the backbone key renders the attacker able to eavesdrop on all user and management traffic. Also, an attacker in possession of the backbone key can insert bogus traffic into the network and thereby congest the network.

In addition, an attacker can make use of the simplified mobility support in MobiSEC and clone a compromised MR and add it to the network in several positions. Finally, the multi-hop property of WMNs is not considered during the authentication protocol. Using the MR as authenticator however means that the key server in IEEE 802.11i will transfer keying material to the MR during authentication. As the MR and the key server do not share a key for this purpose, this key transfer can only be protected with the backbone key and is therefore accessible by any MR in between the MR and the key server.

With respect to MRs joining the backbone a problem will arise if the MR’s certificate is lost and implanted on an attacker’s MR since it is not bound to additional identification attributes.

Multiple operators are not explicitly addressed. Considering the possible communication patterns, Table 3.2 provides an overview with respect to the security requirements that are met by this proposal. The author’s focus is on network access control for MCs along with an additional secondary step for MRs joining the backbone.
3.6. Security Architectures for WMNs

3.6.4 A PANA-based Security Architecture for WMNs

Cheikhrouhou et al. propose the use of the Protocol for Carrying Authentication and Network Access (PANA) [Par05] in the WMN context to carry authentication and key establishment messages during access control of MCs to the WMN [CLMC06]. PANA is an IP-based, medium independent protocol used for carrying other authentication protocols such as EAP. It distinguishes between PANA clients (PaC), agents (PAA), enforcement points (EP), and authentication servers (AS). The authors of [CLMC06] map these PANA entities to the WMN context by considering MCs to be PaCs, EPs to be MRs, PAAs to be MGs, and AS to be a AAA server reachable via each MGs.

Upon a PaC (MC) acquiring an IP address, it receives PAA-Discover messages, which point to the reachable PAAs (MGs). EPs (suggested to be collocated with MRs) are responsible for blocking all non-PANA traffic generated by yet unauthenticated MCs. The authentication phase involves the PaC communicating with the PAA (suggested to be collocated with the MGs) instead of the AS directly. EAP messages are encapsulated in PANA messages exchanged with the PAA. The PAA forwards the de-capsulated EAP messages to the AS which in return provides the material to derive IKE [Kau10] credentials that will be used between PaC (MC) and MR.

Different EAP methods can be used in this context, the authors propose the use of EAP Transport Layer Security (EAP-TLS). After successful authentication the PaC starts an IKE connection with the MR yielding an IPsec tunnel to secure the user traffic. Note that MCs may either be wirelessly connected to a MR directly or to a MR via multiple hops of routing MCs.

Analysis

The authors focus on providing MC access control to the WMN while it is not considered for infrastructure nodes. Authentication and key establishment between MCs and MRs are supported and their communication is protected with the help of an IPsec tunnel. As opposed to this, protecting the communication patterns MC↔MG, MR ↔MR, MR ↔MG and MC↔MC is not addressed. Although multiple operators are not explicitly addressed in the proposal the use of EAP in combination with a hierarchy of AAA servers could allow for the support of roaming MCs. Infrastructure sharing is not addressed in the PANA-based approach.

With respect to supporting routing MCs, the PANA-based approach seems to consider it. However, from the description provided in the paper it remains unclear how routing MCs are supported.
3.6.5 Other Specialized Security Proposals

Some previous WMN security proposals address a single challenge rather than designing a more comprehensive security architecture. We briefly discuss these in this section.

Ren et al. proposed to use a matrix-based key pre-distribution for enterprise WMNs [RL08] that allows two neighboring nodes to establish a pairwise key without exchanging messages. The suggested approach is similar to the multi-map proposal of Gong and Wheeler [GW90]. However, it additionally assumes a mechanism to securely discover locations. The proposal allows for keying material to be established between MCs and MRs.

Islam et al. set out to secure the routing protocol proposed for IEEE 802.11s [IYHH08]. Their proposal uses hash trees to provide integrity protection for the mutable fields of the routing header. The added value in contrast to simple hop-based hashes is questionable. Other security requirements such as confidentiality and privacy are not addressed, but rather referred to as covered by the IEEE 802.11s standard itself.

Kandikattu et al. propose an IBC-based Mobile IPv6 security protocol to enable secure roaming between different trust domains [KJ08]. The secure binding between entities and identifiers is set aside, as well as the roaming between trust domains with different IBC domain parameters. Confidentiality and integrity are achieved by using IPsec between the network nodes. Privacy issues are covered with respect to the location of the home domain.

Hamid et al. [HIH08] propose using modified group signatures developed by Boneh [BF01] to achieve revocability. Users are private within a group, i.e., clients associated with a specific MR. Authentication to a group should be based on in-person contact. They only consider MC↔MR and MC↔MC communication with respect to access control and do not address the other security requirements mentioned in Section 3.5.

As for IBC approaches He et al. [HA10] propose to use IBC for authentication and key establishment on single-hop link layer connections. This effectively competes with the standardized IEEE 4-way handshake [IEE04a] that is already successfully used today in the IEEE 802.11i amendment [IEE04a]. The work in [ZF06] attempts to map IBC in WMNs in a setup similar to that of the credit card ecosystem by involving brokers, which are similar to a CA in a PKI-based system. Operators trust common brokers, which enables clients subscribed to brokers to roam different networks. Wang et al. [WMLW11] propose to unify identity based encryption and certificate-less signatures in a single public key context. They focus on MCs and MRs intra-domain and inter-domain authentication by leveraging trust be-
3.6. Security Architectures for WMNs

Table 3.2: A summary of the characteristics & the security requirements of the compared architectures.

<table>
<thead>
<tr>
<th></th>
<th>Confidentiality</th>
<th>Integrity &amp; Replay Protection</th>
<th>Access Control</th>
<th>Privacy MCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>[IEE11]</td>
<td>MC ↔ MC</td>
<td>MC ↔ MC</td>
<td>MCs</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>MC ↔ MR</td>
<td>MC ↔ MR</td>
<td>MRs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MC ↔ MG</td>
<td>MC ↔ MG</td>
<td>MGs</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MR ↔ MR</td>
<td>MR ↔ MR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MR ↔ MG</td>
<td>MR ↔ MG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[ZF06]</td>
<td>MC ↔ MC</td>
<td>MC ↔ MR</td>
<td>MCs</td>
<td>+</td>
</tr>
<tr>
<td>[MPC08]</td>
<td>MC ↔ MR</td>
<td>-</td>
<td>MCs</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>MR ↔ MR</td>
<td>-</td>
<td>MRs</td>
<td></td>
</tr>
<tr>
<td>[CLMC06]</td>
<td>MC ↔ MR</td>
<td>MC ↔ MR</td>
<td>MCs</td>
<td>-</td>
</tr>
<tr>
<td>[HA10]</td>
<td>-</td>
<td>MC ↔ MC</td>
<td>MCs</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC ↔ MR</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MR ↔ MR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[WMLW11]</td>
<td>-</td>
<td>MC ↔ MR</td>
<td>MC</td>
<td>+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MC ↔ MC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[SZZF11]</td>
<td>-</td>
<td>MCs ↔ MR</td>
<td>MCs</td>
<td>+</td>
</tr>
</tbody>
</table>

Table 3.2: A summary of the characteristics & the security requirements of the compared architectures.

between operators of different domains. The work of Sun et al. [SZZF11] leverages the ideas of [ZF06] in a combination with hierarchical IBC to allow inter-domain authentication (roaming) of clients in FNs.

3.6.6 Discussion

Table 3.2 summarizes the characteristics and high-level security requirements supported by the four previously suggested approaches [HDM+10, CLMC06, MPC08, ZF06]. Note that the summary only shows characteristics and requirements that are met by at least one of the approaches such that some characteristics and security requirements are not included in the table. In particular these are: support of heterogeneous wireless links, mechanisms to enforce fairness, availability protection, and non-repudiation.

From the table it is clearly visible that all the more comprehensive proposals we evaluated in more detail in this section support more than one of the features, while none of them provides a comprehensive security architecture that supports all requirements, not even for a simple single operator WMN. It is important to
note that such a comprehensive security architecture needs to be carefully designed while keeping the possible communication patterns of WMNs in mind.

With respect to protecting the communication between the nodes in WMNs, IEEE 802.11s currently supports all the necessary communication patterns. However, governing the initial network access control by a password makes the standard inflexible, of questionable security, and not suited to support MC roaming or infrastructure sharing (cf. Table 3.3). The ARSA security architecture is focused on first-hop network access, and therefore places much emphasis on the MC↔MR communication pattern. ARSA, however, is the only one of the discussed proposals, which explicitly considers multiple operators and roaming MCs with additional identity privacy. MobiSEC also considers the backbone communication of WMNs, namely that of MRs, and MGs. The authors’ focus is on securing the first-hop communication which also involves access control. Besides controlling MC network access, they also restrict access to the backbone for MRs. MRs have to carry out a secondary authentication after having gained network access with the same mechanism as MCs. The PANA-based security architecture also focuses on network access control and securing the first-hop communication between MC and MR. Nonetheless the first hop is secured by IPsec, the transferral of key material is not secured since other communication patterns such as MR ↔MR are not considered.

### 3.7 Summary & Conclusion

This chapter presented an analysis of WMNs characteristics, security requirements, and possible communication patterns that are affected by these. By stepping back we achieved a broad overview over the security issues WMNs impose as a whole which can differ depending on the different characteristics of the network in ques-

<table>
<thead>
<tr>
<th></th>
<th>Clients</th>
<th>Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>[IEE11]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[ZF06]</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>[MPC08]</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>[CLMC06]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[HA10]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[WMLW11]</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>[SZZF11]</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 3.3: The mobility functionalities of the compared architectures.*
tion. After describing WMNs on an abstract level, and defining the attacker model, we detailed single-operator and multi-operator scenarios whereas the latter is of particular importance (cf. Chapter 6). Also, we defined the most basic communication patterns of WMNs that will be secured by the single operator security architecture proposed in Chapter 5. Additionally, we shortly discussed the concept of mobility for end-users and parts of the network infrastructure. Handover as a mechanism to facilitate seamless mobility will be addressed in Chapter 7. We used our findings of characteristics and requirements to analyze recent security concepts for WMNs. Our analysis shows that the proposed architectures tend to solve only a subset of the problems that we identified.
Part II

The IT Security WMN Testbed
In recent years many testbeds for Wireless Mesh Networks (WMNs) have been implemented to test and evaluate different aspects of WMNs such as the behavior of routing protocols or the performance of Transport Control Protocol (TCP) over multiple wireless hops [ZGW+07, LRBR+12]. None of these is designed with testing and evaluating security mechanisms for WMNs in mind. At the same time, security protocols have been proposed that aim at protecting WMNs [BSH06, CLMC06, ZF06, IYHH08, MPC08, RL08, HIH08, KJ08, BD09, RYLZ10, HA10, WMLW11, SZZF11], covering different aspects of WMN security. However, none of these have been evaluated in a real-world testbed, which makes it very hard to assess the practical use of these proposals let alone to fairly compare them with each other.

We dedicated our new testbed to testing and evaluating security protocols in a realistic setting. This chapter contributes to closing this gap by sharing our experience in designing a testbed dedicated to testing and evaluating security protocols in a realistic setting. In particular, we detail the hardware and software setup of our testbed, describe the management tools we developed to facilitate maintenance of our testbed, and describe several pitfalls we encountered during the setup of our testbed. We also include a general evaluation of throughput, packet loss, and round trip time in the testbed.

Parts of this chapter have been published in [EHJM13].

Outline: This chapter introduces the IT Security Wireless Mesh Network Testbed. For the purpose of implementing and evaluating the key management solutions presented in Chapter 5 and Chapter 6, as well as the handover mechanism (Chapter 7), we designed and created a WMN testbed at the IT Security Research institute.

4.1 ITsec Testbed

This section introduces our WMN testbed, the ITsec Testbed. First, we elaborate on why we chose to build a new custom setup instead of building on the work of others, e.g., the UMIC-Mesh or KAUMesh (cf. Section 4.4.1). Next, Section 4.1.4 details the system image and additional software components used in our testbed.
4.1.1 General Design Considerations

We analyzed a variety of existing testbeds which have been used in non-security specific publicized research (cf. Section 4.4) as there are no security specific WMN testbeds, yet. All of these have been created for specific purposes, e.g., researching routing algorithms, or increasing the performance of transport protocols. As such, they have specific setup and a range of tools and functionalities. For us, the primary requirements were the following: We need to be able to fully control the network topology and the respective routing protocols. This is particularly important for researching handover protocols. For instance, the routing protocol B.A.T.M.A.N.\(^1\) now includes extensions to signal handover from one Access Point (AP) to the next. Other routing protocols without this feature would therefore negatively impact the performance of handover protocols as the network layer session cannot be resumed. Full control over the mode IEEE 802.11 Wireless LAN (WLAN) is operating in, i.e., devices being in station and master mode as opposed to all nodes running in ad-hoc mode. This has significant advantages as the full potential of the 802.11i security mechanisms (cf. Section 2.3) can be leveraged.

We also derived secondary requirements for our testbed: Complete physical control over the nodes in an easy manner. Thus, sharing a testbed with other researchers across countries was not an attractive concept, especially when crashing the nodes. No scheduling of slots for using the testbed. This would otherwise create a lot of unnecessary overhead while limiting the pace of process, e.g., when evaluating the performance of network based mechanisms. This has proven to be of great relevance, especially for students working towards their Bachelor or Master theses. As other testbeds, using Linux on the nodes is superior to any other choice, as it allows the necessary software to be written in most common programming languages.

Hardware components are also similar in all testbeds we analyzed. Most devices are equipped with Atheros chips for wireless connections and have sufficient RAM and computation capacity. A manual installation of the operating systems on each node is obviously not efficient. A comfortable approach to flash multiple nodes with system images is a combination of a Trivial File Transfer Protocol (TFTP) [Sol92] server and a DHCP server. Before booting from the hard disk, the nodes try to get a DHCP lease from the Ethernet interface and load a small system image into their RAM from the TFTP server. This temporary system contains tools to mount a network share containing the desired node system image and to flash it onto the nodes’ persistent storage. Nodes should easily be able to chose whether to boot from a hard disk or their network interface. After the process of flashing the node,

\(^1\)http://www.open-mesh.org/projects/open-mesh/wiki
some settings as setting IP addresses and routing information must be changed to make it work with the testbed.

Hostapd\(^2\) and wpa_supplicant\(^3\) are the standard software for wireless connections and are used in all analyzed testbeds with infrastructure mode. Remotely managing configurations and controlling these daemons is a necessity and should be possible in a convenient way. It also has to be considered that the nodes may differ in hardware. Different hostapd and wpa_supplicant configuration files must be applicable to different interfaces on various nodes.

We also need to be able to manage the Remote Dial-in User Service (RADIUS) server of hostapd as it is required for authentication of all nodes. As opposed the de-facto standard RADIUS server, freeRADIUS\(^4\) does not provide a management interface, yet. Different nodes in a WMN testbed often have similar configurations and only differ in small details, e.g. wireless authentication credentials. We need to be able to provide node-profiles or similar functionality to assign settings to multiple nodes.

### 4.1.2 Architecture

A simplified view of the testbed architecture is shown in Figure 4.1. The network consists of Mesh Routers (MRs), Mesh Clients (MCs), Mesh Gateways (MGs), and a central management server (meshctrl). The mesh router and clients are interconnected

---

\(^2\)http://hostap.epitest.fi/hostapd/
\(^3\)http://hostap.epitest.fi/wpa_supplicant/
\(^4\)http://freeradius.org/
using infrastructure mode. MGs are responsible for routing the traffic to other domains, e.g., the Internet, using a wired backbone. Since each node is authenticated based on Extensible Authentication Protocol (EAP), an Authentication, Authorization and Accounting (AAA) server is necessary. The meshctrl server is connected to the mesh network by wire and implements the RADIUS server included in hostapd to authenticate MRs, MCs and MGs. Also, the management web interface of pwrmesh (cf. Section 4.1.5) is hosted on this server.

4.1.3 Hardware

All nodes run on the PC Engines ALIX.3D3\(^5\) (cf. Figure 4.2) boards which are equipped with a 500 MHz AMD Geode LX800 CPU, an on-chip 128 bit AES Security Block, and 256 MB DDR DRAM. Persistent storage is realized by using 16 GB Compact Flash cards which can be plugged into a CF card slot. The boards also provide two USB ports, a serial port and VGA output. Thus, convenient debugging of the nodes is even possible in case of network failure.

An on-board Ethernet interface with POE capability allows to run the node with a single cable plugged in and with a 100 Mbit Ethernet connection. Two miniPCI sockets are used with two Atheros AR5008 WLAN Cards. Multiple Input Multiple Output (MIMO) technology can be achieved by adding three antennas to each card. The authentication server and all MGs are connected over a Gigabit Ethernet switch.

\(^5\)http://www.pcengines.ch/
4.1. System Image and Software

Voyage Linux\(^6\) is running as an operating system on all nodes, except for the meshctrl server which runs vanilla Debian 6.0. Voyage is a modified Debian Linux with optimizations for wireless drivers, the CF card and other hardware related issues. CF cards have very limited numbers of read and write cycles. Hence, the read and write actions must be reduced as much as possible. Voyage implements a temporary file system which writes each change to the file system into the RAM instead of the CF card. Only defined paths and files are written to the RAM. All other files are mounted read-only. This ensures an increased lifetime of the CF cards and thus also for the devices.

The routers are currently running a 3.2.9 Linux kernel. Important enabled kernel modules are B.A.T.M.A.N. and the i2C module which allows to read values from the on-board temperature sensors. B.A.T.M.A.N. is a proactive routing protocol and is used on all nodes. Communication with the kernel module is possible via the tool `batctl`. It allows to retrieve routing information, e.g., detected gateways, neighbors. All interfaces that use B.A.T.M.A.N. for routing are bridged to a `bat`-device. `Batctl` can also specify whether a device collects or sends visualization data.

Hostapd and `wpa_supplicant` are used for setting up the access points and the wireless connections to the mesh network. In a regular configuration of a node one of the WLAN cards is always connecting to another MR with `wpa_supplicant` and the other is offering an entry access point for other MRs by running a `hostapd` daemon. `Hostapd` can offer multiple virtual interfaces on one physical device. This allows to use one WLAN card not only as an entry point for other MRs, but also for MCs at the same time.

The two Atheros WLAN cards in each node are used with the `ath9k`\(^7\) driver. For higher data rates and less interference with other wireless access points the testbed uses the 5 GHz band. Legacy access for 802.11b/g clients is possible too, but the mesh network traffic is sent using 802.11a.

4.1.5 Node Management

The process of running, maintaining, and configuring the ITsec testbed initially required a lot of manual effort. Additionally, monitoring the testbed was not possible at the time, rendering efficient maintenance even more difficult. In an evolving process, we developed a tailored tool called `pwrmesh`, which uses `pwrcall`\(^8\) [Sch11], a lightweight, secure bidirectional remote procedure call framework as a basis. The

---

\(^6\)http://linux.voyage.hk/  
\(^7\)http://linuxwireless.org/en/users/Drivers/ath9k  
\(^8\)https://github.com/rep/pwrcall
initial capability to which can be accessed by the meshctrl is sent to the MRs during their EAP authentication using the Attribute Value Pairs (AVPs) of the EAP-Tunneled Transport Layer Security (EAP-TTLS) method.

For management purposes we require to be able to change WiFi settings, switch regular nodes to gateway modus, reset/reboot nodes, configure networks settings (IP, iptables, . . . ), and add/remove users. Querying information about the nodes is important to be able to debug and restructure the testbed. Therefore, we obtain the node’s network state, e.g., IP, MAC, connected STAs, and its connectivity. Lastly, we generate the network map using the batmand visualization and merge the information with our local static network topology map. Respective node maintenance we are able to flash new system images, run checks on the CF card and the node’s memory, and reconfigure the PXE-Boot parameters.

We integrated all the above features using an agent-like setup. Figure 4.3 shows the pwrmesh architecture. Each node runs the pwrnode which implements specific functions. All nodes are centrally managed by the pwrserver which is hosted on the meshctrl host. We created a Django\(^9\) based web interface for controlling the pwrserver. Besides automatically rendering the information obtained from each pwrnode, we can also manually trigger commands on the nodes, e.g., rebooting from another PXE-Boot image, or setting up different wireless connections. Figure 4.4 shows an excerpt of the Node View which for instance provides information about the current IP, heartbeat and location. Deploying patches for the most important software, i.e., hostapd, wpa_supplicant, and pwrnode, can also conveniently be

---

\(^9\)https://www.djangoproject.com/
4.2. Pitfalls

In this section we elaborate on the pitfalls we encountered while developing the ITsec testbed.

In the process of using the \texttt{wpa\_wired} driver of \texttt{wpa\_supplicant} we encountered issues with Cisco’s Catalyst 4500 network switch. The MGs of our testbed send EAP requests using the \texttt{wired} driver of \texttt{wpa\_supplicant} as multicast packets to the authentication server. However, the switch drops these packets, even though no filtering whatsoever is employed. Replacing the Cisco hardware with a simple off-the-shelf 1000 Mbit switch resolved this issue.

Additional modifications have been applied to the operating system of the nodes. The directory \texttt{/var/log/} is specified as writable to allow accessing the log files. When log files become too large, the RAM of the devices tends to be completely used for the temporary file system. Hence, we added a script to ensure that no log
Also, there exist issues related to time synchronization of the hardware and the system clock on Voyage. The real time clock cannot be accessed by default, which results in an incorrect system time on startup. This leads to unsuccessful authentications as a correct system time is required for certificate validation of the authentication server. In order to fix this problem the parameter `directisa` needs to be added to each `hwclock` function call and the kernel module `rtc` must be loaded at boot time. Another challenge was the time skew between the node’s clocks which was already too high after a few minutes for any meaningful evaluation. Regularly executing Network Time Protocol (NTP) before each measurement is required.

As to the wireless drivers, some limitations exist when applying multiple virtual interfaces. It is, for instance, not possible to assign different channels for each virtual interface, yet. `hostapd` may also cause problems if the daemon is attached to a device with a specific type of MAC address. In order to distinguish virtual interfaces from the physical device, new MAC addresses need to be assigned. Therefore, `hostapd` chooses the original MAC address as a mask and each virtual interface is configured with this mask increased by its index. The configuration fails if the result of `new_mac & mask` is not equal to the `new_mac`.

When we executed experiments which induced high throughput, `hostapd` tended to terminate the connection resulting in a “deauth” debug message. The station cannot reconnect until `hostapd` is restarted.

Also, there may occur problems to set 5GHz channels on the Atheros cards since most WLAN chipsets have a fixed regional setting. In order to use all legal channels, we patched the `ath9k` driver source code and the regulatory domain
settings. Furthermore, the wireless drivers are far from stable\textsuperscript{10}. If the network traffic is too high over a period of time, or a node loses its connection and then repeatedly scanned for another MR, the driver crashes with a \textit{Direct Memory Access (DMA) error}. Switching to Voyage Linux 0.9 reduced the DMA errors.

Compiling and building an application using Linux is straightforward. There are several tools which assist developers in the process. However, compiling and installing freeRADIUS is not as easy as one might expect. It is important to use the correct version of \textit{libtool}\textsuperscript{11}. For us, this means completely removing the installed libtool version and use the one which is included in freeRADIUS. Unfortunately freeRADIUS does not provide a de-installation script. Therefore, we wrote a custom removal script for freeRADIUS such that we could make sure that only the current version was used.

4.3 Performance Evaluation

This section presents the network related performance metrics.

4.3.1 Network Performance

In order to obtain the most important performance metrics of the ITsec testbed, we carried out measurements for packet loss, \textit{Round Trip Time (RTT)}, and throughput. All the measurements have been done using \textit{iperf} and have been repeated a significant amount of times to obtain stable results. The parameter \(n\) refers to the number of repetitions and \(t\) refers to the duration of a specific test instance.

\textsuperscript{10}http://wireless.kernel.org/en/users/Drivers/ath9k/bugs

\textsuperscript{11}http://www.gnu.org/software/libtool/
Figure 4.7: The average round trip times in two different setups.

Figure 4.8: The average throughput of the ITsec testbed.

Figure 4.6 shows the average packet loss of User Datagram Protocol (UDP) transmissions. We have used an additional 5 MBit UDP noise stream on the same path, however, it only slightly influences the average packet loss.

In terms of RTT, we compared a close to optimal Line of Sight (LOS) setup with a setup in which the routers were regularly distributed throughout our institute. Figure 4.7 shows that the difference between both scenarios is almost negligible. Obviously, RTTs increase as the overall distance increases.

For the purpose of determining the throughput we ran a 20 seconds test using TCP, i.e., measuring the available bandwidth using a constant stream for 20 seconds. Figure 4.8 shows the comparison of the two scenarios as discussed before. However, the optimal LOS setup produces a significantly larger throughput.
4.4 Related Work

This section summarizes the most relevant set of related testbeds which we considered during our work. We focused on these non-security specific testbeds, as there are no security specific WMN testbeds, yet. A comparison respective testbed features and usage is also given in Table 4.1.

4.4.1 UMIC-Mesh

The UMIC-Mesh [ZGW+07] is a hybrid testbed which consists of physical devices and virtualized nodes. Each node is connected to a wired backbone network that is used for configuration and booting. This functionality is supported by a configuration server which is also connected to the wired backbone. The mesh internal traffic is routed over wireless connections. The physical devices are running on ALIX.2C2/3C2 boards from PC Engines. Atheros AR5213 XR WLAN cards are used for wireless communication. One card is dedicated to router-to-router communication whereas the other is dedicated to router-to-client communication. The routers are connected in 802.11 ad-hoc mode using OLSR [CJ03] as the default routing protocol. The UMIC-Mesh is configurable and monitored using the web-based tool MeshConf which is based on SNMP [Pre02]. Each node being connected to the wired backbone makes it easy to configure, however, it also renders the deployment scenario less close to real world mesh networks. The main research focus of the UMIC-Mesh is multi-hop performance of transport protocols such as TCP.

4.4.2 KAUMesh

KAUMesh12 is a testbed developed at the Karlstad University in Sweden. Similar to the UMIC-Mesh, all nodes are connected to a management and monitoring server by wire, while wireless connections are established using 802.11 ad-hoc mode. The nodes are built using the Cambria GW2358-4 network computer with Intel IXP435 667 MHz CPUs and 128 Megabyte DDR2 SDRAM running the Linux operating system. Nodes operate up to three Atheros wireless interfaces. Monitoring the nodes is based on Nagios13 using Simple Network Management Protocol (SNMP). Each node retrieves its system image via Trivial File Transfer Protocol (TFTP) [Sol92]. There is no configuration tool to configure the nodes directly as the necessary configuration is done at boot time. Three different routing protocols are available in the testbed: AODV-UU [PBRD03], B.A.T.M.A.N., and OLSR [CJ03]. Each node is equipped with

---

12https://www.kau.se/en/kaumesh
13http://www.nagios.org/
several monitoring tools to retrieve data, e.g., *iperf, tcpdump, mgen*. The KAUMesh is used for performance and QoS studies.

### 4.4.3 UCSB MeshNet

The UCSB MeshNet [LRBR+12] is a testbed developed at the University of California, Santa Barbara. It consists of 30 nodes deployed in a single building connected in ad-hoc mode. Each node consists of two Linksys WRT54G wireless devices which are connected via Ethernet. One WRT54G is responsible for the mesh traffic. The other device tries to connect to other access points in order to get an Internet connection or another connection that is available to a testbed operator. The gateways of the networks are Intel Celeron based PCs which connect via IEEE 802.11b to the mesh network and provide Internet access via Ethernet. Both node types are running Linux based operating systems, i.e., the Linksys nodes are running OpenWrt14. The testbed uses *Ad-hoc On Demand Distance Vector (AODV)* [PBRD03] for routing and the researchers developed their own control software for it but there is no detailed description or source code available. The UCSB MeshNet is used for routing and QoS research.

### 4.4.4 DES-Testbed

The Distributed Embedded Systems Testbed15 is developed by the Freie Universität Berlin. It is a hybrid mesh testbed containing 95 nodes. Each node takes part in the wireless mesh network and also in a wireless sensor network. Some of the nodes are connected to the wired backbone of the network. A controlling testbed server is also connected to this wired backbone. The nodes are based on PC Engines ALIX.2C2/2D2/3D2 system boards with three or more 802.11a/b/g Atheros wireless network cards communicating in IEEE 802.11 ad-hoc mode and one sensor node. These system boards are capable of running Linux operating systems. The testbed is managed by a custom tool called DES-TBMS and used for routing and localization studies.

### 4.4.5 Freifunk

Freifunk16 networks are private non-commercially deployed mesh networks. Participants can share their Internet connection with other Freifunk members or can enlarge the mesh by connecting to an existing node while acting as a forwarder. In

---

14https://openwrt.org/
15http://www.des-testbed.net/
16http://start.freifunk.net/
order to participate in the Freifunk network, a special firmware based on OpenWrt
exists for the routers which can be deployed on a variety of off-the-shelf hardware.
The firmware provides OLSR and B.A.T.M.A.N. as routing protocols. Wireless
connections between nodes are established in ad-hoc mode. Freifunk networks are
community networks which do not have a specific academic research focus.

4.5 Summary & Conclusion

In this chapter we presented the ITsec Testbed as an experimentation platform for
WMN research. We detailed the initial construction and our specific requirements
towards a testbed for security research. The security architecture which is presented
in the following chapter is the cornerstone of the research that follows. Also, the
subsequent Chapter 5-Chapter 6 show research that has been sparked by the simple
fact of a testbed being available. For instance, handover protocols for WMNs, have
to the best of our knowledge not been implemented and evaluated using a WMN
testbed, yet. Our approach to a testbed shows that using off-the-shelf components
facilitates building a testbed which enables researchers and students to obtain real
world practical results.
<table>
<thead>
<tr>
<th>Testbed</th>
<th>Mode</th>
<th>OS</th>
<th>Security</th>
<th>Hardware</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMIC-Mesh</td>
<td>Ad-hoc, 2x WLAN</td>
<td>Linux</td>
<td>N/A</td>
<td>ALIX.2D2 500 MHz, 256 MB RAM</td>
<td>TCP Performance</td>
</tr>
<tr>
<td>KAUMesh</td>
<td>Ad-hoc, 3x WLAN</td>
<td>Linux</td>
<td>N/A</td>
<td>Cambria GW2358-4, 667 MHz, 128 MB RAM</td>
<td>QoS, Performance</td>
</tr>
<tr>
<td>UCSB MeshNet</td>
<td>Ad-hoc, 2x WLAN</td>
<td>OpenWrt</td>
<td>N/A</td>
<td>Linksys WRT54G, 200 MHz, 16 MB RAM</td>
<td>Routing, QoS</td>
</tr>
<tr>
<td>DES</td>
<td>Ad-hoc</td>
<td>Linux</td>
<td>N/A</td>
<td>ALIX.2C2/2D2/3D2 500 MHz, 256 MB RAM</td>
<td>Routing, Localization</td>
</tr>
<tr>
<td>Freifunk</td>
<td>Ad-hoc</td>
<td>OpenWrt</td>
<td>N/A</td>
<td>Many models possible</td>
<td>Community Network</td>
</tr>
<tr>
<td>ITsec Testbed</td>
<td>Infrastructure, 2x WLAN, Gateways wired</td>
<td>Linux (Voyage)</td>
<td>802.11i, WPA EAP, FSASD</td>
<td>ALIX.3D3 500 MHz, 256 MB RAM</td>
<td>Security Research</td>
</tr>
</tbody>
</table>

**Table 4.1:** This table compares different WMN testbeds on a general level.
Part III

A Comprehensive Security Architecture for Multi-Operator WMNs
This chapter proposes a comprehensive framework for securing wireless mesh networks that is fully compatible to the IEEE (IEEE) 802.11s standard [IEE11]. The framework enables the efficient establishment of all security associations required for end-to-end protection of the different traffic types in the Wireless Mesh Network (WMN).

As any wireless network, WMNs are particularly vulnerable to external attackers trying to eavesdrop on or manipulate traffic sent over the wireless links. They might also try to gain unauthorized network access. However, the multi-hop nature of WMNs combined with the potentially exposed placement of Mesh Routers (MRs) and with the fact that Mesh Clients (MCs) may route traffic for other MCs, induces additional security challenges for WMNs. Specifically, 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 in the WMN must be ensured. Once malicious devices have been identified, there needs to be a mechanism to remove them from the network.

Other previous approaches (e.g., [MPC08] or those we discussed in Chapter 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 are not compatible to the IEEE 802.11s [IEE11] standard and therefore only stand a slight chance of getting widely used within a commercial context. Another weakness of previously suggested key management frameworks as well as industrial solutions such as MONTOMESH [Mot06] is that they are based on proprietary or at least non-standardized protocols.

In this chapter, we address the WMN specific security challenges described above by proposing a comprehensive security framework which

(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 multi-hop traffic;

(5) supports secure proactive handover of moving MCs (or MRs) from one point of network attachment to another.

Our proposed framework is fully compatible to the IEEE 802.11s [IEEE11] amendment and can be realized with commercially available OTS devices. We implemented the framework in our live WMN testbed described in Chapter 4 and evaluated the performance of its components.

Parts of the work presented in this chapter have also been published in [EFM12].

Outline: We start by introducing a three party protocol in Section 5.1 which we use as an idea for our own three party protocol. Next, in Section 5.2 we detail our framework, including the network assumptions, security requirements, key derivation and multi-operator considerations, a three party handshake protocol, and the network deployment phases. After that, we introduce the FSASD Authenticator (FAU) in Section 5.3 as a logical component that is used to speed up our key management protocols by leveraging the hierarchical structure of WMNs. We follow with a comprehensive security analysis of all the framework components. Before concluding this chapter in Section 5.6, we present a performance analysis of our framework in Section 5.5.

5.1 Preliminaries

In this section we briefly describe a 3-party key transport protocol of Marin-Lopez et al. [LPGBHGS10]. It 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. We chose this protocol as a basis for the three party handshake protocol presented in Section 5.2 as it is already embedded into the context of wireless networks using Extensible Authentication Protocol (EAP). The protocol consists of the following messages:

Starting with $M_1$, $A$ initiates the protocol with $B$. $M_1$ includes its identifier $A$ and a token encrypted by $A$ for $S$ with the symmetric key $K^\text{auth}_{AS}$ shared between $A$
Figure 5.1: The message flow of the 3PFH protocol.

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 sends a token for A (encrypted with \( K_{AS}^{auth} \)) and a token for B (encrypted with \( K_{BS}^{auth} \)) to B. A’s token contains the identities of both A and B, as well as 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 (Section 5.2.5). We chose this protocol for two main reasons: First, it has been proposed in the context of handover in Wireless LANs (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 Automated Validation of Internet Security Protocols and Applications (AVISPA) [Vig06], a widely accepted formal tool for protocol verification. As will be shown in Section 5.4, our variant of the protocol still positively passes the evaluation with AVISPA.

Note that we explicitly decided not to use 802.11r [IEE08] for two reasons: First, 802.11r assumes that access points already share security associations, however, without defining how these are to be established. Second, 802.11r assumes a 802.11 MAC layer for wireless connections, and Ethernet for wired connections. Therefore, WMNs using heterogeneous wireless access technologies cannot be supported. Our
solutions presented in this thesis are PHY and MAC layer agnostic, thus supporting a wide range of technologies.

5.2 Framework for Establishing Security Associations in Sequential Deployment

In this section we detail the Framework for establishing Security Associations for Sequentially Deployed WMN (FSASD) security framework. We start by discussing what the framework assumes as the network model. After that, we introduce the security requirements that FSASD aims to fulfill, and which technical means and protocols are used.

5.2.1 Network Assumptions

We assume that the WMN (i.e., all MRs and Mesh Gateways (MGs)) is operated by a single operator who 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 Authentication, Authorization and Accounting (AAA) server present in the WMN. This AAA server may, but does not have to be, co-located with a MG. Each MG, MR, and MC shares authentication credentials, e.g., user name and password, with the AAA server. 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 true for user traffic routed through the WMN to and from the Internet.

5.2.2 Security Requirements

In Chapter 3 we introduced general security requirements for WMNs which we used to evaluate different security architectures for WMNs and wireless ad-hoc networks.

This section instantiates these requirements in the context of a comprehensive security architecture for WMNs. The framework presented in this chapter intends to fulfill these derived requirements.

Due to their multi-hop nature, WMNs are particularly vulnerable to active and passive external attackers on the wireless links. They 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, the MC’s traffic to and from the MG has to be protected against compromised MRs and routing MCs. Similarly, authentication traffic between a Network Access Server (NAS) and the AAA server is at risk. Thus, it has to be protected against compromised MRs and routing MCs.

In accordance with the analysis presented Chapter 3 we derive 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.

5.2.3 Overview

Our framework meets all the requirements listed in Section 5.2.2 and in particular solves the problem of bootstrapping the necessary security associations. Our proposal is fully compatible with the IEEE 802.11s standard, PHY and MAC layer agnostic, 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 N1 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 Master Session Key (MSK) and the Extended Master Session Key (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 Master Session 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 based on the consecutive keys on each wireless link.

The second key - the EMSK - is used as root in a hierarchy of keys illustrated in Figure 5.2. From the EMSK we derive the Root Traffic Key (rTK) from which an Internet Protocol Security (IPsec) security association (containing a Traffic Encryption Key (TEK) and a Traffic Integrity Key (TIK)) is derived. If later \(N_1\) acts as NAS, these keys are used to protect the authentication traffic between \(N_1\) and the AAA server by using IPsec (R4).

The two remaining keys Root Peer Authentication Key (rPAK) and Root Key Derive Key (rKDK) of the key hierarchy are used for authentication and key derivation during bootstrapping the security associations required to meet R5, R6, and R7. For this bootstrapping we propose the 3-Party Handshake for Sequential Deployment (3PHSD) as detailed in Section 5.2.5. 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 rKDK and the AAA server securely transfers this key to node B by leveraging the IPsec security association that has been bootstrapped during its authentication.

Note that our framework concentrates on protecting the communication within the WMN. Protecting traffic that leaves the WMN via a MG and is destined to the Internet or other networks is out of scope of our framework. The MG that is used for Internet connectivity can also listen on outgoing communication if the traffic is not secured otherwise, e.g., by Transport Layer Security (TLS) to a host on the Internet. This is not a severe restriction, since MRs and MGs are devices of the network operator, which are expected to behave correctly. Of course, this assumption becomes obsolete if such a device is captured or compromised by an
5.2.4 Key Derivation

The root of the key hierarchy is the EMSK, a key of at least 64 bytes in length which must be exported by any key-generating EAP method [ASE08]. The keys that are subsequently derived from the EMSK are cryptographically separated. In the following we describe the key derivation process and length of the derived keys (cf. Table 5.1).

We use the *Key Derivation Function (KDF)* PRF+ specified in RFC 5996 [Kau10], which can be based on any keyed cryptographic hash function. In our framework 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 (cf. Equation 5.1). This KDF suits our goal of key derivation from the EMSK best, since it generates 8160 bytes of cryptographically independent key material. It is efficiently computable and can be implemented for use with different cryptographic hash functions. Moreover, it is possible to use additional input and salt in addition to the EMSK.

\[
\text{KEY} = \text{PRF(EMSK||Name||0x00||Salt||Length)}
\] (5.1)

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 and TIK are both 256 bits long. The rPAK, the rKDK and the MSK-L1 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, rKDK, 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 rKDK, it may have a lifetime shorter than the lifetime of the EMSK if rKDK’s lifetime is shorter.

**PAKID**

We now introduce the so-called *Peer Authentication Key ID (PAKID)*. It is a temporary identifier linked to the rPAK of a node and it is used by our protocols, e.g., 3PHSD. The purpose of the PAKID is to preserve the identity privacy of MCs. Instead of using the MAC address or other long-term identifiers, we use the PAKID.

It is used by the AAA server to map a specific rPAK to the respective node. This mapping is necessary since the AAA server needs to select the correct rPAK to
<table>
<thead>
<tr>
<th>Key Type</th>
<th>Key Length</th>
<th>Key Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Master Session Key (EMSK)</td>
<td>⩾64 bytes</td>
<td>The root of the key hierarchy is exported by key-generating EAP methods and is only available at the EAP server and EAP peer.</td>
</tr>
<tr>
<td>Root Traffic Key (rPAK)</td>
<td>64 bytes</td>
<td>Derives traffic protection keys.</td>
</tr>
<tr>
<td>Traffic Encryption Key (TEK)</td>
<td>⩾128 bits, default 256 bits for AES-CBC</td>
<td>Bootstraps confidentiality between future NAS devices and authentication server.</td>
</tr>
<tr>
<td>Traffic Integrity Key (TIK)</td>
<td>⩾256 bits, default 256 bits for HMAC-SHA-256</td>
<td>Bootstraps integrity and authentication between future NAS devices and authentication server.</td>
</tr>
<tr>
<td>Root Peer Authentication Key (rPAK)</td>
<td>64 bytes</td>
<td>Authenticares the token of a peer in a three-party protocol.</td>
</tr>
<tr>
<td>Key Derive Key (rKDK)</td>
<td>64 bytes</td>
<td>Derives fresh keys for the EAP client and the new authenticator.</td>
</tr>
<tr>
<td>Master Session Key Level 1 (MSK-L1)</td>
<td>64 bytes</td>
<td>Provides a fresh MSK to the parties in 3PHSD.</td>
</tr>
</tbody>
</table>

*Table 5.1:* The various key types, key lengths & purposes used in FSASD.
5.2. Framework for Establishing Security Associations in Sequential Deployment

authenticate and en-/decrypt protocols messages.

\[
\text{PAKID} = \text{HMAC-SHA1-256}(r\text{PAK}, \text{“PAK Name”}|\text{User-Name}|\text{“auth”}) \quad (5.2)
\]

Equation 5.2 defines the generation of the PAKID. The rPAK shared between any node and the AAA server is used to key the SHA1-based Keyed-Hash Message Authentication Code (HMAC). Additionally, a label is supplied along with the long term user name of the node and the use case of the ID. As such, the PAKID serves as a temporary Unique Identifier (UID) during the protocol runs of 3PHSD in single- and multi-operator scenarios. Thus, intermediate, possibly untrusted or compromised nodes cannot deduce long term identities of other nodes by overhearing the PAKID. The lifetime of the PAKID is limited by the lifetime of the rPAK.

Multi-Operator Considerations

The introduction of the generic key hierarchy easily allows to include a multi-operator concept. Keys that are generated in a specific WMN or a sub-domain, are Domain Specific Root Keys (DSRKs) according to RFC 5295 [SDNN08]. They must only be used in a specific domain of either a network or a specific operator. To support roaming and authentication delegation a Domain Specific Key Hierarchy may be used. In this case, so called Domain Specific Usage Specific Root Keys (DSUSRKs) are derived from a DSRK. How we extend the FSASD concept to a scenario supporting multiple operators will be shown in detail in Chapter 6.

5.2.5 3-Party Handshake for Sequential Deployment

This section describes the 3PHSD protocol. 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 (SA) between MC and MG (to meet R5).

The performance evaluation of the protocol can be found in Section 5.5.3.

In accordance to Section 5.2.3 we use the following notations:

- A, B, S : Identity of Peer A, Peer B, and AAA Server S
- rPAK\(_{AS}\) : Peer Authentication Key between A and S
- MSK-L1 : Resulting pairwise key between A and B
- \(\{x\}_k : x\) encrypted by key \(k\)
- \(N_A, N_B, N_S\) : Nonce of A, B, and S
A 3PHSD protocol run consists of four messages shown in Figure 5.3. Message M1 contains the identity of Peer A and Token 1 authenticated and encrypted by the $\text{rPAK}_{AS}$ 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 5.2.6). Server S can authenticate and decrypt Token 1 based on $\text{rPAK}_{AS}$. Using a key derivation function with the $\text{rKDK}$ 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 B (cf. Equation 5.3).

In Message M3, S directly sends the MSK-L1 to Peer B, along with Token 2 authenticated and encrypted by $\text{rPAK}_{AS}$. 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
5.2. Framework for Establishing Security Associations in Sequential Deployment

Token 2 and use its contents to generate the MSK-L1 using the rKDK as key input and the nonces and a key label 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.

**Differences to 3PFH.** Relying on FSASD, i.e., the key hierarchy and the IPsec SAs that are automatically bootstrapped during network deployment, our protocol cannot only be used for handover as originally proposed, but also in a more general sense, i.e., bootstrap a pairwise key between arbitrary authenticated nodes with the help of our key hierarchy.

3PHSD only uses encrypted tokens for the messages that are sent between the initiating peer and the AAA-server. Other messages, i.e., *Remote Dial-in User Service (RADIUS)* messages, are encrypted and integrity protected by the IPsec SA that is already in place (cf. Section 5.2.3). The token of the first peer and the RADIUS message are bound together by the integrity algorithm of IPsec. EAP messages in 3PHSD are *not* limited to the 802.11 MAC layer as in the original proposal by Marin-Lopez et al. [MLOPG10], but may also be sent on top of an IP protocol representing an alternative EAP lower layer, e.g., *Transport Control Protocol (TCP)*, *stream control transport Protocol (SCTP)*, or *User Datagram Protocol (UDP)*.

We envision two ways 3PHSD can be initiated, the first one being at the time of wireless association of a new device, e.g., MC or MR. The second way is to send 3PHSD messages via an alternative EAP lower layer to another peer, which becomes necessary if the participants are not in direct radio range. For this purpose we implemented two alternative EAP lower layers on UDP and SCTP (cf. Section 5.5). This is of particular relevance in bootstrapping security associations between MCs and MGs. Since traffic of MCs will typically leave the WMN via the MG, the traffic traversing multiple hops has to be sufficiently secured from attacks of intermediate nodes. Therefore, 3PHSD is used to bootstrap a shared symmetric key between a MC and a MG, which can be used by IPsec. This makes an additional *Internet Key Exchange Protocol (IKE)* handshake obsolete, as the key can be directly used after 3PHSD completes. It is possible to use 3PHSD as the MG must be authenticated before it can act as such, thus yielding the proposed key hierarchy. MCs running 3PHSD with the MG are also already authenticated, thus the MCs are also in possession of the proposed key hierarchy.

In addition, using 3PHSD along with FSASD conveniently enables to dynamically integrate more MGs into the WMN. MCs are only required to discover newly added MGs in the WMN, however, this is out of scope of this work.
5.2.6 Network Deployment

Our framework assumes a sequentially deployed WMN, i.e., nodes are dynamically added to the network while some are already present. In this section we describe how the different types of nodes are deployed. This section elaborates on the resulting network deployment phases that have been divided into three phases: Gateway-, Mesh Router-, and Mesh Client-Phase.

Mesh Gateway Deployment

In the gateway deployment phase, the AAA server and the MG are set up. We differentiate two scenarios, namely the MG being co-located with the AAA server and 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.

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 5.2.3) 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., $TEK_{MR_1}$, $TIK_{MR_1}$, $rTK_{MR_1}$, $rPAK_{MR_1}$, and $rKDK_{MR_1}$. 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 IEEE 802.11 4-way handshake succeeded, link layer encryption and integrity protection 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 over multiple wireless hops from $MR_1$ to AAA.

Mesh Client Deployment

The client deployment is similar to that of routers, 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
5.3 FSASD-Authenticator

As the performance evaluation in Section 5.5 shows, the distance between the joining node and the node which is responsible for authentication and authorization is the decisive performance factor. In the case of EAP-Tunneled Transport Layer Security (EAP-TTLS), this is the distance between Peer and AAA server. Considering 3PHSD, we refer to the distance between Peer A and Server S. While performance optimization is not of immediate importance for 3PHSD when being used to establish security association between an MC and the MG, it matters significantly in the handover scenarios. If the distance between the MC and the AAA server can be reduced, the overall runtime of the protocol will decrease.

For this purpose we propose the so called FAU. It is a new logical component whose purpose it is to reduce the distance between the parties of the protocols and the AAA server. The FAU has similar capabilities as the AAA server, i.e., it can generate and distribute keys for specific use cases. In a WMN with an inherent hierarchical structure several FAUs may exist, thus, dividing the network into logical sub-domains.

As Figure 5.4 shows, the AAA server represents the top level of the hierarchy of FAUs. The other FAUs are co-located with MRs. It must be ensured that such an MR provides the same security properties as the AAA server since it stores keying material for many nodes. A FAU has two tasks:

1. Storing duplicates of the keys of the FSASD key hierarchy

2. distributing subsequently derived keys, e.g., Pairwise Master Keys (PMKs) in a handover context, to the corresponding target node.

FAUs may be in possession of keys that were generated by the AAA server. If a FAU is co-located with an MR it needs to obtain the keys from the AAA server.
over the network to the FAU_{MR} in a secure fashion. Recall that each node in the network was authenticated using an EAP method which derived the FSASD key hierarchy (cf. Figure 5.2). Thus, the communication between the FAU_{MR} and the AAA server is secured by IPsec using the TIK and the TEK. The rPAK shared between the FAU_{MR} and the AAA server could also be used to securely transport messages between both. Based on this security association, secure key distribution to the FAU_{MR} is possible.

Each FAU_{MR} manages only the keys needed by the nodes in its sub-domain. If the FAU_{MR} is currently not in possession of a required key, it forwards the request to the AAA server which possesses the necessary keys. Thus, it is possible to define a minimal set of keys necessary at the FAU to enable specific services.

### 5.3.1 FAU Key Hierarchy Extension

In order to fully exploit the potential of the extensible key hierarchy of FSASD, we now extend it to enable FAU support. Figure 5.5 shows the extended key hierarchy. As previously explained, the EMSK is the root of the hierarchy. Since introducing the FAUs effectively splits the WMN into several logical sub-domains, we create keys bound to a specific FAU domain.

This means that the SAs a node shares with the AAA server are based on the root keys, i.e., those on the first and second level of the key hierarchy (red). The root Traffic Key (rTK) is used to derive the TIK and TEK for the IPsec SA between the node and the AAA server. Next, the root Key Derive Key (rKDK) is used to derive further keys, i.e., the FAU Domain Root Keys (FDRKs) which is specific to a FAU domain. The root Peer Authentication Key (rPAK) is used for bootstrapping of further security association which involves the node and the AAA server.
Each FDRK represents the root of a key hierarchy which is specific to the current FAU domain such that FDRKi \neq FDRKj for different FAUs i \neq j. Its key derivation is shown in Equation 5.4. In order to bind the FDRK to a specific FAU during key derivation, we use the fau_id which corresponds to a Network Access Identifier (NAI) and its IP address as other devices need to be able to address the FAU. The FDRK is unique for each (FAU,Sn) tuple as each node has a unique rKDK.

MRs need to be able to bootstrap an IPsec SA to the FAU of the domain they are located in. After their authentication with the AAA server they generate the key hierarchy and derive the corresponding FDRK. Using this key they subsequently derive the root Traffic Key for this FAU domain, rTK_{FDi}, and the respective traffic keys to be used with IPsec (blue). Thus, key transport between FAU and MRs is secured in the same way as key transport between the AAA server and the MRs.

The PMDK_{FDi} (black) is a key which is used to derive further PMKs.

The PAK_{FDi} (green) is the Peer Authentication Key (PAK) to be used in FAU domain i for the same purposes as the rPAK is used in the super-domain, i.e., to bootstrap SAs during which the FAU is involved, or to encrypt and authenticate protocol messages.

### 5.3.2 FAU Bootstrapping & Mobility

We now show how nodes can bootstrap a FAU. This involves determining the current FAU which is responsible for the region the node currently resides in, and
Figure 5.6: The security associations in a FAU-enabled WMN. The colors used correspond to Figure 5.5.

to generate the respective key according to the key hierarchy shown in Section 5.3.1.

Nodes can determine the responsible FAU by overhearing the beacon (cf. Section 2.3) of the MR they currently connect to. Once a node has determined the FAU-ID and the FAU IP address of FAU\(_i\), it can generate the FDRK\(_i\) (cf. Equation 5.4). Additionally, relevant keys can now be derived according to the key hierarchy shown in Section 5.3.1.

FDRKs are pushed to the FAUs by the AAA server once the EAP authentication of a node finishes and also whenever a EAP session times out or is refreshed. The FAU can now derive further keys according to Equation 5.1 by replacing the EMSK with the FDRK. Now the FAUs are ready to bootstrap additional SAs with other nodes using 3PHSD. Depending on the deployment setup, an authenticating node may indicate the responsible FAU to the AAA server during its EAP authentication, e.g., using EAP-TTLS Attribute Value Pairs (AVPs). After that, the AAA server pushes the requested FDRK to the corresponding FAU.

Changing the FAU domain is just as simple as its bootstrapping process. If at some point it becomes necessary for a node to change from the domain of FAU\(_i\) to that of FAU\(_j\), the node needs to bootstrap the FAU of the new domain. The responsible FAU of the other FAU-domain can be recognized by parsing the wireless beacons, the node can simply generate the corresponding FDRK\(_j\) used in
the new domain.

5.4 Security Analysis

This section presents a security analysis of our proposed framework FSASD. We discuss the security requirements and the overall achieved goals, i.e., confidentiality, integrity, authenticity, availability, and replay protection. As Marin-Lopez et al. [LPGBHGS10] we also use AVISPA [Vig06] to formally evaluate the security of the 3-Party Handshake for Sequential Deployment (3PHSD).

We also cross check the recommendations of Request For Comments (RFC) 4962 [HA07] “Guidance for AAA Key Management” against our security architecture and our newly proposed protocols. It thus fits very well to the context of our protocols. The RFC belongs to the “Best Current Practices” category and describes conditions that a AAA protocol or a collection of protocols from which one of them is an AAA protocol should satisfy.

5.4.1 Key Management

Table 5.2 summarizes the security features and mechanisms FSASD provides and how these relate to the security requirements R3-R7 (see below). 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 (while not using FAUs) 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 by IPsec.

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

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 [RS04].

Compromise of session keys does also not compromise future or past session keys (forward secrecy). 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.

5.4.2 FSASD Authenticator

In this section we discuss the security consideration of the FAU nodes.

A FAU can be considered special in the sense that it does store key material not only for itself, but also for other nodes. Thus, the FAU is a mixture between an AAA server and a MR. As we consider the AAA server to be secure and thus, sufficiently hard to compromise, but the MR untrusted and more easy to compromise, the FAU requires a careful analysis.

First of all, we require the FAU to be installed in areas which make it hard to physically compromise the node, especially in contrast to regular MR which may for instance be installed on street lights. The system security of a node acting as a FAU should also be of better quality than that of regular MRs.

The keys stored by FAUs are of short term nature, i.e., they time out and will be refreshed no later than the time out of the EAP session of a node. At this point, the AAA pushes new FDRKs to the FAUs. Key transport between AAA server and FAU is secured by an IPsec SA, which according to the FSASD deployment, has previously been bootstrapped after the authentication of the FAU node itself.

In contrast to the AAA server, important long term identifiers, i.e., user names and passwords used during the EAP authentication are not stored on the FAU.

Compromising a FAU affects only its specific domain as only FAU-domain specific keys are stored.

For instance, impersonation of nodes which are currently requesting a handover is possible, which, however, is the same as for the AAA server. Consider a FAU attempting to impersonate a node that has just requested a handover. For successful impersonation, the FAU needs to change its MAC address to the nodes it attempts to impersonate and use the corresponding PMK during the 4-way handshake with the handover target. However, eavesdropping and subsequently breaking confidentiality on the link layer is impossible, since the established handover keys
are explicitly confirmed during the 4-way handshake which in turn establishes new transient session keys to be used on the link layer. For a successful attack, also the 4-way handshake needs to be captured by the attacker.

### 5.4.3 3-Party Handshake

In this section we 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 Encapsulating Security Payload (ESP) connection between the AAA server and B. M2 and M3 are replay protected by IPsec in the same way. The AAA server S can detect a replay of Token 1 (cf. Figure 5.3) 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 \( rPAK_{AS} \). B will always receive the same MSK-L1 from the AAA server S which 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 rKDK. Thus, 3PHSD meets the same security goals as the 3-party protocol originally proposed by Marin-Lopez et al. [LPGBHGS10] 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 slightly delays the detection of replay as it is shifted to another host.

We also performed a formal analysis of 3PHSD using AVISPA [Vig06], an automatic protocol verification tool. 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 [LPGBHGS10] 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 token to S. In our 3PHSD this encryption and
Comm. Pattern | SA bootstrapped by | Secured by
--- | --- | ---
single-hop (R3) | EAP based on AAA credentials | CCMP using keys derived from MSK
MR↔MC, MG, MR

single/multi-hop (R4) | EAP based on AAA credentials | IPsec using TEK and TIK
NAS↔AAA

multi-hop (R5,R6) | 3PHSD based on rPAK | IPsec using keys derived from MSK-L1, derived from rKDK
MC↔MC, MG, MR

Broadcast (R3) | EAP based on AAA credentials | CCMP using GMSK derived from MSK
MC, MG, MR →*

handover (R7) | 3PHSD based on rPAK | CCMP using keys derived from MSK-L1, derived from rKDK
MC↔new NAS

Table 5.2: Protected communication patterns in a WMN secured by FSASD.

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.

5.4.4 Summary

In terms of RFC 4962 [HA07], “Guidance for AAA Key Management”, we draw the following conclusion:

- **Cryptographic algorithm independence** is achieved as we rely on EAP and only require the EAP method to be key generating, which does not depend on specific cryptographic algorithms. This is also true for the IPsec connections between MRs/MGs and the AAA server. Algorithm independence can also be achieved for 3PHSD by replacing encryption and hashing mechanisms.

- **Strong fresh session keys** are provided by EAP and specifically by our extensible key hierarchy. The keys are bound to the lifetime of the EAP session and
Security Analysis

are cryptographically separated. Brute force attacks on any key generated by the hierarchy or 3PHSD are infeasible as all keys are of sufficient length.

Key scope is defined for all generated keys, either those generated by EAP, as well as all keys generated by our extensions. For instance, the rTK is specifically used to generate the rTEK and rTIK used for the IPsec connections between MRs/MGs and the AAA server.

Replay protection for key delivery is achieved by EAP itself, as well as by IPsec.

Authentication and authorization of all parties is based on the EAP method. As for establishing the IPsec connections, the involved parties are implicitly authenticated by the key used on this connection. Non-authenticated parties cannot send valid packets. After authentication, all involved parties are implicitly authorized by key possession. In 3PHSD, the communication parties are implicitly authenticated based on the keys using for computing the ciphertext of the sent packets. Also, key confirmation is implicitly achieved after the protocol finishes.

Confidentiality and integrity protection of key material is ensured by the secure key transport via IPsec between MRs/MGs/MCs and the AAA server.

Confirming ciphersuits is implicitly built into EAP and IPsec. 3PHSD does currently not support ciphersuite negotiation.

Uniquely named keys are achieved for all keys in our key hierarchy.

Prevention of the domino effect is relevant for the FAUs introduced in Section 5.3. Compromising the AAA server obviously compromises all keys. If a FAU is compromised, the impact is localized to the managed sub-domain of the FAU. The key in question can only be misused for the lifetime of the EAP session. Only temporary key material is stored on the FAU, but no long term credentials. Thus, compromising one FAU neither compromises other FAUs, nor does it compromise the AAA-server.

Binding the keys to their context is provided in our key hierarchy by using a specific key label during key generation.

As can be seen by the security analysis, FSASD offers comprehensive security and satisfies important goals for AAA based key management solutions. Table 5.2 summarizes the security features of our framework.
5.5 Implementation & 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 [Pos80] and SCTP [OY02].

5.5.1 Implementation

Our proposed security framework is implemented using our testbed described in Chapter 4. The necessary key derivation functionality has been implemented as a patch to the freeRADIUS AAA server. Client side changes to derive the FSASD key hierarchy and subsequently using them for, e.g., IPsec and 3PHSD are integrated into the de-facto standard WLAN Station (STA) and Access Point (AP) software wpa_supplicant and hostapd.

5.5.2 Performance of EAP Authentication

Figure 5.7 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 a black bar. Each measurement is labeled with the median and minimum and maximum are marked by whisker-bars.
5.5. Implementation & Evaluation

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

5.5.3 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 lower layers: SCTP and UDP to transport EAP messages. Both protocols are simple packet-oriented rather than stream-oriented as TCP. This allows us to efficiently encapsulate 3PHSD messages into SCTP or UDP messages. The time was measured by the peer initiating the 3PHSD starting from sending of the first message, computing and displaying the new MSK until receiving the fourth message.

The tuples on the horizontal axis of Figure 5.8 represent the distances between Peer A and Peer B, as well as the distances between Peer B and the Server S. For instance, 2-3 refers to the distance between Peer A and Peer B being two hops, and the distance between Peer B and the AAA server being three hops.

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

In the second scenario (2), Peer A represents an MC that 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. This best corresponds to the tuple 5-2, i.e., 5 hops from the MC to the MG and 2 hops from the MG to the AAA server.

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 a 3PHSD protocol run. However, larger distances between Peer A and Peer B 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 as 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.
Using it 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 than 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.

5.6 Summary & Conclusion

In this chapter we presented FSASD, a novel comprehensive 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 standard IEEE 802.11s. As opposed to the standard, our proposal reduces the influence of compromised MRs on the overall security of the network to a minimum and protects MCs from curious and malicious routing MCs. In addition, our framework enables secure and efficient handover of MCs and MRs based on the new 3PHSD protocol. This protocol is later (cf. Chapter 7) used to develop and extend our architecture to enable an even more efficient and practical handover solution. By using the concept of FAUs we are able to leverage the hierarchical structure of WMNs, thereby, improving the key management capabilities especially for handover purposes. Our framework complies to IEEE 802.11s amendment and it 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 se-
curity features our framework provides is negligible and that framework is thus well-suited for direct practical use.
WMNs are often considered to be operated by a single operator which owns the complete infrastructure nodes and offers connectivity to MCs that are registered with it. There are, however, a variety of application scenarios that profit from or even require multi-operator support. The most obvious is the support of roaming clients as known from mobile telephony networks. Here an MC is not only able to use the WMN operated by his Home Operator (HO), i.e., the operator it registered with, but also the WMN of any other Foreign Operator (FO) that has a roaming agreement with its HO. In addition, there are application scenarios in which the infrastructure nodes, namely the MRs and MGs, forming the WMN are operated by different operators. An example for such an application is building automation using WMN technology in which different companies may be responsible for specific parts or functions of the building. Also, community networks in which users contribute their own network equipment require this form of multi-operator support. Finally, disaster recovery scenarios can highly profit from multi-operator support using a common mixed infrastructure formed by the network equipment of different first responder units.

Roaming security in WMNs is quite similar to other wireless networks and mainly involves the support of authentication and key agreement across different operators. Adequately protecting mixed infrastructure networks is, however, quite challenging. In particular, it requires procedures for the secure deployment of infrastructure nodes across different operators. In addition, mixed infrastructure networks imply that nodes deployed in the same WMN can differ greatly with respect to their resilience against attacks. In particular, MRs and routing MCs should not be considered uncompromisable and trustworthy.

In this chapter we propose a comprehensive security architecture for multi-operator WMNs supporting roaming clients as well as mixed infrastructures. Our proposal is a generalization of the single-operator solution proposed in Chapter 5. However, we considerably extend this proposal to securely support roaming MCs and secure the deployment of infrastructure nodes across different operators. Our solution is based on open standards and protects traffic generated by MCs from insider attackers such as compromised MRs, MRs operated by malicious operators,
and curious or malicious routing MCs.

The work presented in this chapter has also been published in [EM13].

Outline: After summarizing the related work on multi-operator concepts in Section 6.1, we continue in Section 6.2 by discussing our network architecture, security requirements, and the necessary foundations. After introducing our approach in Section 6.3 and evaluating its security in Section 6.4, we follow with a performance discussion in Section 6.5 and conclude in Section 6.6.

6.1 Related Work

Research with specific focus on multi-operator WMNs is rare. In general we can observe two distinct trends. Identity-based Cryptography (IBC) [BF01] on the one hand has been a popular concept that is used by many researchers [ZF06, HA10, SZZF11, WMW12] mostly leveraging the fact of self-authenticating public keys. On the other hand research [ASE08, BD09] is more focused on multi-domain networks using standardized protocols.

As for IBC approaches, He et al. [HA10] propose to use IBC for authentication and key establishment on single hop link layer connections. This effectively competes with the standardized IEEE 4-way handshake [IEE04a] that is already successfully used today in IEEE 802.11i [IEE04a]. The authors argue that running EAP [ASE08] between an MC and the AAA server is too complex and has too much communication overhead for fast session refresh or handover. The author’s simulations result in a “constant small value” for the distance of a single wireless hop which is of course also true for the IEEE 4-way handshake. The work in [ZF06] attempts to map IBC in WMNs in a setup similar to that of the credit card ecosystem by involving brokers, which are similar to a Certification Authority (CA) in a Public Key Infrastructure (PKI)-based system. Operators trust common brokers, which enables clients subscribed to brokers to roam different networks. Wang et al. [WMLW11] propose to unify identity based encryption and certificate-less signatures in a single public key context. They focus on MCs and MRs intra-domain and inter-domain authentication by leveraging trust between operators of different domains. The work of Sun et al. [SZZF11] leverages the ideas of [ZF06] in a combination with hierarchical IBC to allow inter-domain authentication (roaming) of clients in Foreign Networks (FNs).

One of the most prominent approaches addressing multi-operator WMNs using standardized mechanisms and protocols is the work of Buttyán et al. [BD09]. The authors focus on MC re-authentication, i.e., a subsequent authentication process
after a full-fledged authentication run, by delegating the authentication from the AAA server to the APs. Considering non-trusted MRs, or those placed in physically insecure locations, authentication delegation can create serious threats for network operators. Their evaluation shows that running the protocol and its variants on the single hop connection between MC and AP is obviously faster than a full EAP authentication run as no interaction with the AAA server is required. On the other hand our prior work presented in Chapter 5 provides the groundwork for the research put forth in this chapter. It does so by building on top of the medium independent EAP protocol suite and by leveraging mechanisms such as IPsec.

In general, the IBC-based mechanisms suffer from the inherent problems of IBC such as key expiration, non-revocable public keys, the challenge of secure private key generation and key distribution. As such, securely initializing the various domains and in turn delivering the parameters to the network devices remains an unsolved problem. Thus, [ZF06, HA10, SZZF11] and [WMW12] inherit a bootstrapping-problem.

We solely rely on EAP for all devices, thus, there is no explicit differentiation between MCs, MRs, and MGs during authentication. We use well scrutinized security protocols which are available in current OTS hardware today. Using our WMN testbed [EHJM13] proves that the often voiced communication overhead is less than expected, even for complex EAP methods such as EAP-TTLS. By leveraging EAP our architecture does not suffer from the bootstrapping problem, as all generated domain specific keys have the purpose to be used to bootstrap additional SAs, which would otherwise suffer from this very problem. Any devices can be authenticated via other previously authenticated devices, regardless of them belonging to the same management domain or them being of another operator. Even handover for MCs as well as for infrastructure components such as MRs is possible while roaming in an FN.

6.2 System Model and Foundations

This section discusses the network assumptions and model. We also introduce the necessary key management and key derivation concepts, i.e., domain specific keys, which are required to understand our proposal.

6.2.1 Network Model

In the following we distinguish two multi-operator scenarios, namely classic separate networks as in most roaming scenarios. Additionally, we consider mixed networks
which are a novel concept that has to the best of our knowledge not been considered thus far in any prior research in this area.

**Separated Networks**

In multi-operator networks which are fully separate, a single network consists strictly of infrastructure devices of one specific operator. More specifically, infrastructure devices such as MRs, MGs, the AAA server of each network are not necessarily the property of the operator. Additionally, in both of the scenarios MGs, MRs, and MCs may serve as the point of network attachment, i.e., NAS, for newly joining MCs or MRs. Figure 6.1 shows an example on the left side. The network is operated by the operator **Red** which maintains the MRs, MGs and the AAA server which may be co-located with a MG. A MC of the operator **Blue** is currently roaming in this network.

**Mixed Networks**

In so-called mixed networks, devices of one operator may not only connect to devices of the same operator, but also to devices of other operators. We assume a network consisting of infrastructure devices of multiple operators, either private and/or commercial. Each operator is assumed to be running his own AAA server, e.g., at its **Home Network (HN)**. While considering mixed networks, the notions of HN and FN become somewhat fuzzy. In contrast to a HN which is maintained by the MC’s home operator, the FN is maintained by a different operator. Therefore, we define a HN of a device as the network in which its AAA is operated in, even though the network itself may consist of devices from multiple operators. Thus, for an infrastructure device being in a FN means being connected to a network where the **local** AAA server is not operated by its HO.

Furthermore, we assume MGs not to be co-located with the AAA server, except for MGs of the local domain. Deploying a AAA server in a FN is not considered as they store sensitive user data and credentials onto which operators typically enforce restrictive policies. Figure 6.1 depicts an example of a mixed multi-operator scenario on the right side. Devices of different operators (**Green, Red**) are connected to each other while the blue operator is responsible for running the local AAA server.

### 6.2.2 Multi-Operator Security Requirements

The wireless multi-hop nature makes WMNs particularly vulnerable to active and passive external attackers on the wireless links. Attackers may try to gain unauthorized network access or may try to eavesdrop on or manipulate the traffic
in the WMNs (cf. Section 3.3). Moreover, we consider MRs and MCs, which may also route traffic, to be untrusted. These devices may be placed in easily accessible areas and can therefore be compromised. Consequently, these compromised MRs and routing MCs may try to eavesdrop on and manipulate the traffic passing through them. Therefore, MC traffic to and from MGs must be end-to-end protected.

In order to secure multi-hop traffic of MCs (e.g., towards the Internet), an IPsec security association can be bootstrapped with a MG. Securing multi-hop client traffic in roaming scenarios in networks with multiple operators can be differentiated into two sub-scenarios. Securing the traffic between the roaming MC and the MG of its HN (cf. Section 6.3.2), and securing the traffic between the MC and the MG of the currently visited FN (cf. Section 6.3.2). Nonetheless, insider threats are relevant for both scenarios, mixed networks still pose an increased threat as devices from different operators may not be equally trustworthy. It may for instance be desirable to secure MC traffic by a VPN-like setup to the MC’s HN instead of using the MG of the FN if it is not fully trusted.

The same holds true for authentication traffic between a NAS and the AAA server. It is especially important for MCs that MRs which relay their authentication traffic to the AAA server have a secure connection. Also, once the MC or MR has been authenticated, a secure multi-hop connection must be established as the MC may not trust the local operator, especially not the MRs which may be placed in physically insecure locations.
Besides secure authentication and access control for all devices (MRs, MGs and MCs), hop-by-hop link layer security is required to prevent outside attacks such as eavesdropping and traffic manipulation. This leads to the security requirements as presented in Section 5.2.2.

To meet R1 and R2 a protocol for mutual authentication between joining nodes and the AAA server is required (e.g., EAP), as well as a mechanism to exclude compromised nodes from the network. In order to meet the requirements R3-R7 security associations between the communicating parties must be bootstrapped.

As we will see in this chapter, all these security requirements can be achieved by extending our single-operator security architecture, which we presented in Chapter 5.

6.2.3 Domain Specific Keys

Domain Specific Keys are an important concept when discussing EAP-based multi-operator networks. These keys are specific to a domain, i.e., they are generated to be used in a specific network domain. When an explicit usage for a key is specified, e.g., to derive a handover key, it is referred to as a Usage Specific Root Key (USRK) according to RFC 5295 [SDNN08]. They must only be used in the mesh network in the domain of a specific operator. To support roaming and authentication delegation in other domains, a so-called Domain Specific Key Hierarchy can be used. In this case, Domain Specific Usage Specific Root Key (DSUSRK) are derived from a Domain Specific Root Key (DSRK). RFC 5295 discusses how to derive these keys from the EMSK.

According to the document, DSRKs or DSUSRKs can "be made available to and used within specific key management domains" [SDNN08], i.e., from the AAA server of one domain to the AAA server of another domain. However, the (secure) key transport between AAA servers (e.g., RADIUS) has not been specified, yet. Recently Hoeper et al. [HNO10] proposed a method to transport keys used by the EAP re-Authentication Protocol (ERP) [ZHZ12] (which are DSUSRKs) between different key management domains. The authors discuss message transport and that it needs to be secure (confidential, authentic, and integrity protected), however, how the respective security mechanism are to be bootstrapped is not specified.

Depending on the trust relationship between the domains, the exporting domain may also chose only to export the DSUSRKs. This effectively limits the allowed applications of these keys in the FN. However, in the following we assume that a full domain specific key hierarchy is exported to the FN. This theoretically enables the FN to generate additional DSUSRKs for specific application scenarios in its network. As such, the trade-off between flexibility and trust needs to be carefully
6.3. Approach

Now, we introduce our multi-operator key hierarchy extension allowing secure deployment of all infrastructure nodes as well as secure and efficient roaming of MCs. The key hierarchy introduced in Chapter 5 can be extended such that multi-operator scenarios are supported without sacrificing the original security properties. In the following we use \( \langle \text{device} \rangle_{\text{location}}^{\text{allegiance}} \) to express operator allegiance of a device and its current location. For instance, \( \text{MR}_\psi^\pi \) refers to a MR that belongs to Operator \( \psi \) and is currently located in the network of Operator \( \pi \). \( \langle \text{key} \rangle_x \) refers to a domain specific key to be used in the Domain \( x \), e.g., \( \text{rPAK}^\pi \) refers to the \( \text{rPAK} \) of a device that can be used in the Domain \( \pi \).

### 6.3.1 Multi-Operator Key Hierarchy Extension

In *Framework for establishing Security Associations for Sequentially Deployed WMNs (FSASD)* (cf. Chapter 5) the EMSK is a Root Key. Subsequently derived USRKs such as the \( \text{rPAK} \) are used to bootstrap other security mechanisms. As such, the EMSK is also an USRK, as its sole purpose is to derive the keys on the next level in the key hierarchy which themselves have a specific purpose.

If we now extend this notion from one single domain to multiple domains, we need to introduce so-called *Domain Specific Keys*. As introduced in Section 6.2.3, these keys are used and valid only in domains specific to their definition. For instance,
when the key hierarchy is generated for the domain of Operator \( \psi \), a so-called DSRK needs to be derived from the EMSK. The DSRK\( ^{\psi} \) (cf. Figure 6.2) now represents the root of the FSASD key hierarchy for domain \( \psi \). Subsequently derived keys using DSRK\( ^{\psi} \) are defined for a specific usage and the domain \( \psi \), i.e., they are so-called DSUSRKs.

The FSASD key hierarchy has been generated by the AAA server of a domain the authenticating device belongs to. It can for instance export DSRK\( ^{\psi} \) to Operator \( \psi \), and DSRK\( ^{\pi} \) to Operator \( \pi \). In their respective domains, these operators are now able to generate the Domain Specific Usage Specific Root Keys (DSUSRKs) for their key management domain (cf. Figure 6.2). The mechanisms that we now propose in the following assume that the full domain specific key hierarchy is exported to the FN.

### 6.3.2 Operator-separated Networks

In the following we assume that the network of each operator is deployed according to the FSASD security architecture as introduced in Chapter 5. Therefore, all authentication traffic in each network is confidential, authenticated and integrity protected. Operators must have a bilateral service agreement in order to allow for roaming of clients from different operators in their own network. In the next section we discuss the use case of a client roaming from its HO \( \psi \) to a FO \( \pi \). Deploying MCs, MRs and MGs of the same operator works according to the single-operator FSASD approach presented in Chapter 5.

#### Client Roaming

A client in allegiance to Operator \( \psi \), MC\( ^{\psi} \), is roaming in a FN of Operator \( \pi \). We denote such a client as MC\( ^{\psi}_{\pi} \). When the client first connects to a NAS (the MR it associates to) of the foreign domain it initiates a regular EAP authentication. Recall that we assume the network of both domains \( \psi \) and \( \pi \) to be deployed using FSASD, i.e., most importantly the authentication traffic is secured by IPsec from the NAS to the AAA server.

A client of Operator \( \psi \) connecting in the FN \( \pi \), i.e., MC\( ^{\psi}_{\pi} \) involves the following steps (cf. Figure 6.3):

1. MC\( ^{\psi}_{\pi} \) associates to NAS\( ^{\pi}_{\pi} \) which relays the authentication requests to the RADIUS server, AAA\( ^{\pi}_{\pi} \).

2. AAA\( ^{\pi}_{\pi} \) recognizes the request being a roaming device of Operator \( \psi \) and proxies the message [RRSW97] to AAA\( ^{\psi}_{\psi} \) in the MC's HN.
6.3. Approach

Figure 6.3: Authentication of MC\textsubscript{ψ} which is roaming in the FN of Operator π.

(3) AAA\textsubscript{ψ} authenticates MC\textsubscript{ψ}, derives FSASD domain specific keys and relays the response to AAA\textsubscript{π}. This also includes exporting the DSRK\textsuperscript{π} of MC\textsubscript{ψ} to AAA\textsubscript{π}.

(5) AAA\textsubscript{π} relays the response to NAS\textsubscript{π} enabling its mutual authentication between MC\textsubscript{ψ} and NAS\textsubscript{π} based on the IEEE 802.11i 4-way handshake.

As the connecting device does not belong to the local domain, the AAA server of Operator π will proxy the EAP authentication request to the devices’ Home Network (HN), i.e., the AAA server of Operator ψ. The decision to which domain the request needs to be forwarded is based on the NAI [ABE05]. Forwarding the request from the AAA of one domain to the AAA of another domain must use a secure connection.

The key transport between RADIUS servers has not been specified, yet. We suggest to use the message specification of Hoeper et al. [HNO10] as neither [RRSW97], nor [SDNN08] elaborates on secure key transport other than stating that it must be secure. If the bilateral service agreement between operators includes securing the connection between their AAA servers (e.g., by using IPsec), we consider this issue as resolved. However, if this is not the case, we propose that the MGs of cooperating operators are authenticated by the respective AAA of the other operators in the
context of FSASD. As a consequence, the MG of Operator $\psi$ would be able to bootstrap an IPsec connection with the AAA server of Operator $\pi$ based on the TEK$^\pi$, and TIK$^\pi$ (cf. Figure 6.2) and vice versa. This would resolve the issue of secure communication between the AAA servers of different operators. The MG is either assumed to be co-located with a local AAA server, or having been authenticated by the AAA server and thus having a secure channel to it.

Depending on the trust relationship between the domains, the home AAA server can choose whether to export the complete domain specific key hierarchy to the AAA of the other domain, or only export DSUSRKs such as the rPAK. As a result of this process, $MC^\psi_\pi$ has been authenticated in cooperation between Operator $\psi$ and $\pi$. All keys were transported securely whether using wired or wireless technology, and the link layer security between $MC^\psi_\pi$ and $NAS^\pi_\pi$ has been enabled.

**Securing MC Multi-Hop Traffic:** $MC^\psi_\pi \leftrightarrow MG^\psi_\psi$

Suppose the HN is controlled by Operator $\psi$ and a client $MC^\psi_\pi$ is currently roaming in the FN of Operator $\pi$. After the client has successfully been deployed using the mechanism described in Section 6.3.2, $MC^\psi_\pi$ can start 3PHSD (cf. Section 5.2.5) with the MG of its HO, i.e., $MG^\psi_\psi$. Recall that each device the WMN of Operator $\psi$ has been authenticated using our security framework FSASD and is in possession of the FSASD key hierarchy.

Also, we rely on the PAKID (cf. Section 5.2.4) as a temporary identifier to allow the AAA server to identify the messages and to avoid sending long term identifiers.
6.3. Approach

across the network. This is especially relevant when roaming in an FN.

The message exchange is depicted in Figure 6.4.

\[ M_1: PAKID_{MC}^{\psi} (N_{MC}^{\pi}, t_{MC}^{\psi}, MG_{\psi}^{\pi})_{rPAK^{\psi}} \]

\[ M_2: M_1||MG_{\psi}^{\pi}, N_{MC}^{\psi} \]

\[ M_3: \{N_{MC}^{\psi}, N_{AAA}^{\psi}, N_{MG}^{\psi}, MG_{\psi}^{\psi}, AAA_{\psi}^{\psi}\}_{rPAK^{\psi}}, K \]

\[ M_4: \{N_{MC}^{\psi}, N_{AAA}^{\psi}, N_{MC}^{\psi}, MG_{\psi}^{\psi}, AAA_{\psi}^{\psi}\}_{rPAK^{\psi}} \]

The first message \( M_1 \) sent from \( MC^{\psi} \) to \( MG^{\psi} \) contains a nonce and a timestamp of the client and the identity of \( MG^{\psi} \). This message is encrypted and integrity protected by \( rPAK^{\psi} \) which is a DSUSRK for domain \( \pi \) and shared between \( MC^{\psi} \) and \( AAA^{\psi} \). When \( MG^{\psi} \) receives \( M_1 \), it generates and appends a nonce and its identity and transfers it via the secure connection to \( AAA^{\psi} \) as message \( M_2 \). Recall that the MG is either co-located with the AAA, or has been authenticated by the AAA, thus having bootstrapped an IPsec connection based on the FSASD key hierarchy. The \( AAA^{\psi} \) now generates the key \( K \) (cf. Equation 6.1), appends it to the parameters for \( MC^{\psi} \), i.e., the nonces of \( MC^{\pi}, AAA^{\psi} \) and \( MG^{\psi} \), which are encrypted and integrity protected by \( rPAK^{\pi} \) and sends it as message \( M_3 \) to \( MG^{\psi} \).

It now relays the secured message to \( MC^{\psi} \) which can now generate \( K \) using the contents of message \( M_4 \).

\[ K = \text{HMAC-SHA1-128}(rKDK^{\psi}, \text{"Home MG"}||N_{MC}^{\psi}||N_{AAA}^{\psi}||N_{MG}^{\psi}) \quad (6.1) \]

This key can now be used to set up a secure multi-hop connection (e.g., an IPsec tunnel) between \( MC^{\psi} \) in the network of Operator \( \pi \) and \( MG^{\psi} \) of its HO. The result of the protocol is a VPN-like IPsec connection from \( MC^{\psi} \) residing in the network of Operator \( \pi \), to the \( MG^{\psi} \) in its home network of Operator \( \psi \). The MC’s traffic is kept confidential from all intermediaries in the FN and between both networks.

**Securing MC Multi-Hop Traffic**: \( MC^{\psi} \leftrightarrow MG_{\pi}^{\pi} \)

Roaming MCs can also use the MG of the FN which they are currently roaming in (cf. Figure 6.5).

The message flow is similar to the case discussed in Section 6.3.2. Recall that once \( MC^{\psi} \) has been authenticated by the network of Operator \( \pi \), the DSRK\(^{\pi} \) of \( MC^{\psi} \) is transferred from \( AAA^{\psi} \) to \( AAA^{\pi} \). As a result, the roaming client now shares a key with the local \( AAA^{\pi} \) of the FN, namely \( rPAK^{\pi} \). This key enables the
Figure 6.5: Bootstrapping an IPsec connection from the roaming MC$^\psi_\pi$ to an MG of the FN.

The execution of 3PHSD with the gateway MG$^\pi_\pi$ of the FN. The first message M1 is secured by encrypting and integrity protecting it using the key rPAK$^\pi$ which is a DSUSRK for the Domain $\pi$. M2 between MG$^\pi_\pi$ and AAA$^\pi_\pi$ is secured by the IPsec security association which has been bootstrapped during the FSASD deployment (if they are not co-located).

M1: PAKID$^\pi_{MC}$, $\{N_{MC}^\psi, t_{MC}^\psi, MG^\pi_\pi, rPAK^\pi\}$

M2: M1||MG$^\pi_\pi$, N$^\pi_{MG}$

M3: $\{N_{MC}^\psi, N_{AAA}^\pi, N_{MG}^\pi, MG^\pi_\pi, AAA^\pi_\pi, rPAK^\pi, K\}$

M4: $\{N_{MC}^\psi, N_{AAA}^\pi, N_{MG}^\pi, MG^\pi_\pi, AAA^\pi_\pi, rPAK^\pi\}$

Message M3 is sent after AAA$^\pi_\pi$ has generated K (cf. Equation 6.2). It sends it to MG$^\pi_\pi$ along with the encrypted and integrity protected parameters for MC$^\psi_\pi$. Once MC$^\psi_\pi$ has received the parameters in message M4 and has generated the key K, an IPsec connection securing the multi-hop traffic between MC$^\psi_\pi$ and MG$^\pi_\pi$ can be bootstrapped based on K.

$$K = \text{HMAC-SHA1-128}(rKDK^\pi, \text{“Foreign MG”} \mid N_{MC}^\psi \mid N_{AAA}^\pi \mid N_{MG}^\pi) \quad (6.2)$$

6.3.3 Mixed-Infrastructure Networks

In the following we assume a FSASD-deployed network of devices of multiple operators. We now discuss the cases of different devices of the domain of Operator $\psi$ connecting to the network.

Connecting Mesh Routers

When an infrastructure device connects to a mixed network, it either associates to a NAS of a different operator, or to one of the same operator. In any case, the
6.3. Approach

![Diagram]

Figure 6.6: Authenticating MR$^{\psi}_{\pi}$ in the domain of Operator $\pi$.

NAS has already been authenticated before, i.e., FSASD security associations can be assumed, especially those securing authentication traffic between NAS and AAA.

However, to which operator the NAS in question belongs to is relevant. Each MR serving as a NAS has an IPsec connection to its HO AAA server securing authentication traffic. For instance, if MR$^{\psi}_{\pi}$ has an IPsec connection to AAA$^{\psi}_{\psi}$, while being connected in the FN domain of Operator $\pi$, authentication traffic for devices belonging to a different operator (e.g., the local Operator $\pi$) must always be relayed through AAA$^{\psi}_{\psi}$. As a result, the authentication traffic of a device MR$^{\psi}_{\pi}$ is delayed by the indirection MR$^{\psi}_{\pi}$ → AAA$^{\psi}_{\psi}$ → AAA$^{\pi}_{\pi}$. We therefore argue that any device, irrelevant of operator allegiance, should bootstrap the IPsec connection securing the authentication traffic with the local AAA server (in the following AAA$^{\pi}_{\pi}$) of the present network. Figure 6.6 shows the steps involved to authenticate MR$^{\psi}_{\pi}$ in the local domain of Operator $\pi$.

On the one hand, authentication requests of devices not belonging to the domain of Operator $\psi$ have to be proxied by AAA$^{\pi}_{\pi}$ to their HO and AAA$^{\psi}_{\psi}$ anyway. On the other hand, authentication requests of devices of Operator $\pi$ are thereby not delayed by traversing through a AAA server of a different operator. In addition, this behavior is standard compliant to RADIUS which only proxies request of devices belonging to domains different from its own domain.

Bootstrapping the IPsec security association is based on the DSRK$^{\pi}$ (and the derived keys TEK$^{\pi}$ and TIK$^{\pi}$ (cf. Figure 6.2)) which AAA$^{\psi}_{\psi}$ exports to AAA$^{\pi}_{\pi}$ for a connecting device MR$^{\psi}_{\pi}$.
Connecting Mesh Gateways

Adding more gateways to a WMN increases the overall bandwidth. Particularly devices for which the hop distance to a possible new gateway is less than the distance to other gateways profit from decreased latency and increased bandwidth. The deployment of a new gateway to a mixed network is very similar to the router deployment discussed in the previous section. For instance, once a gateway, e.g., MG\textsubscript{$\psi$}, has been authenticated in the mixed network of the local domain of Operator $\pi$ by the mechanism described in Section 6.3.3, it can easily be integrated. As such, MGs can bootstrap an IPsec connection based on the DSUSRKs with the local AAA server; just as MRs do.

Recall that DSRKs are transferred from the gateway’s HN to the local FN. MCs, irrelevant of operator allegiance, can thus bootstrap an IPsec connection to MG\textsubscript{$\psi$} in operator domain $\pi$. For instance, 3PHSD messages between MC\textsubscript{$\psi$} and AAA\textsubscript{$\pi$} can be secured using their shared key $rPAK^{\pi}$. The messages between MG\textsubscript{$\psi$} and AAA\textsubscript{$\pi$} are secured by their IPsec connection.

Connecting Mesh Clients

An MC connecting to a mixed network is essentially the same as in the case of separately operated networks. During association it is irrelevant to which operator the NAS has its allegiance to, as the NAS will have an IPsec connection to the AAA server of the local domain. The MC’s authentication traffic is therefore either proxied by the local AAA to its home domain, or the MC belongs to the local domain and its authentication traffic is destined for the local AAA anyway. In either case, the MC generates the FSASD key hierarchy specific to the domain it is currently connected to. If it is a foreign domain, the DSRKs are exported from the MC’s home AAA to the local AAA.

Bootstrapping IPsec to a gateway is achieved by running 3PHSD. As discussed in Section 6.3.3, gateways are authenticated using FSASD and DSRK may be exported from home to local AAA servers. As the connection between MG and AAA servers is assumed to be secure in both co-located and non co-located scenarios, MCs can simply run 3PHSD with MGs regardless of the MG’s operator allegiance. For instance, in the local domain of Operator $\pi$ messages between MC\textsubscript{$\chi$} and AAA\textsubscript{$\pi$} can be encrypted and authenticated based on the shared $rPAK^{\pi}$ and messages between MG\textsubscript{$\mu$} and AAA\textsubscript{$\pi$} based on their IPsec connection.

Figure 6.7 shows the authentication of arbitrary devices in a mixed network scenario in the local domain of Operator $\pi$. All but the requests of devices of Operator $\pi$ have to be proxied to the AAA of the HN of the device.
6.4 Security Considerations

Three important security aspects have to be considered: Deriving and exporting domain specific keys, security of separated networks, and security of mixed networks.

The derivation of the DSRKs should be done as defined in [SDNN08] which also states that DSRKs should not be exported to domains with uncertain authorization, i.e., rather DSUSRKs should be exported selectively.

Regarding transport from one domain to another requires confidential and authentic key transport. In [HNO10] the so-called Key Distribution Exchange (KDE) is proposed which aims to specify the transport of domain specific keys in context of ERP [ZHZ12]. We assume the AAA server and a MG to be either co-located or both being individual entities. To make use of the key hierarchy that is created during device authentication, a device, such as a gateway or the AAA server would have to be authenticated by the operator requiring key transport. The connection MG↔AAA is secured either by both being co-located, or by the gateway having been authenticated by the AAA server. Thus, we achieve authentic, integrity protected and confidential key transport from one domain to another.

In the setting of separated networks we only need to consider roaming MCs, e.g., MC$_H^\psi$ associating to MR$_H^\pi$. Authentication traffic from NAS to the local AAA server is secured by their bilateral IPsec connection. Proxying the MC’s request to the AAA server of its HN is secured by the secure connection between the two operator networks.

Once the DSRK, or selected DSUSRKs have been securely exported to the AAA server of the FN, additional security mechanisms can be bootstrapped. For instance, the MC$_H^\psi$ can choose to either bootstrap an IPsec connection to a MG of its HN, or to a local gateway MG$_H^\pi$ of the FN. Compared to using the MG of the HN a
performance increase will be the result as the traffic does not need to be home-routed as known from some Mobile-IP scenarios [Per10]. However, the MC multi-hop connection to a local MG in an FN primarily safeguards the intermediate hops of the FN, i.e., from insider attackers such as routers and possible other clients of arbitrary operators. Traffic of MC leaving the gateway of the FN is vulnerable to eavesdropping by the FO if no additional security measures such as TLS are used on higher layers. It is thus a trade-off between trust and performance. Link layer communication between MC and NAS is authentic, integrity protected and confidential based on keys derived during the EAP authentication.

In the mixed-network scenario, export of domain specific key material is secured by the same mechanisms as in separated networks. Authentication traffic of MCs, MRs, or MGs with allegiance to arbitrary operators in the local domain of another operator is secured by IPsec between the local AAA server and the NAS the device is connecting from. The operator allegiance of the NAS is irrelevant to the authentication process as well as to the connecting device, as each NAS has an IPsec connection to the local AAA server instead of to the AAA server of the home operator. Traffic initiated by MCs is secured against untrustworthy intermediaries by using IPsec to either the MG of the MCs HO, or the MG operated by the local operator.

6.4.1 Summary

As for the initial single-operator FSASD introduced in Chapter 5, we evaluate our multi-operator solution in terms of RFC4962 [HA07]:

- **Cryptographic algorithm independence** remains the same as in single-operator FSASD. We do not require specific cryptographic algorithms, thus, they can be replaced if needed.

- **Strong fresh session keys** are provided by EAP and specifically our extensible key hierarchy. This is also true for the extension using the concept of DSRKs.

- **Key scope** is defined for all generated keys, either those generated by EAP, as well as all keys generated by our extensions. For instance, the scope of keys is limited to a specific domain and usage (DSUSRKs).

- **Replay protection** for key delivery is achieved by EAP itself, as well as by IPsec.

- **Authentication and authorization of all parties** is based on the EAP method which is used during roaming, mixed and separated networks. No further differences to the single-operator FSASD exist.
6.5 Performance Considerations

- **Confidential and integrity protection** of key material is ensured by the secure key transport via IPsec between MRs/MGs/MCs and the AAA server as in single-operator FSASD, even across domains. Inside a specific domain, the key transport is equivalently secured as in single-operator FSASD.

- **Confirming ciphersuits** is implicitly built into EAP and IPsec. 3PHSD does currently not support ciphersuite negotiation.

- **Uniquely named keys** are achieved for all keys in our key hierarchy. Keys being valid in a specific domain are named accordingly.

- **Prevention of the domino effect** is relevant for the FAUs introduced in Section 5.3. Compromising the AAA server obviously compromises all keys. If a FAU is compromised, the impact is localized to the managed sub-domain of the FAU. The keys in question can only be misused for the lifetime of the EAP session. Only temporary key material is stored on the FAU, but no long term credentials.

- **Binding the keys to their context** is provided in our key hierarchy by using a specific key label during key generation. Also a context is created by introducing the concept of DSRKs.

6.5 Performance Considerations

The performance of FSASD as originally proposed is only minimally affected by generalizing it to a multi-operator concept. In Section 5.5 we evaluated the EAP authentication performance in our real-world IEEE 802.11 testbed (cf. Chapter 4) which included an impact evaluation of using IPsec between the NAS and the AAA server. Using EAP-TTLS over 1-5 hops from the NAS and the AAA server ranges roughly from 380ms to 450ms.

In the infrastructure-separated networks (cf. Section 6.3.2) a delay respective to the latency between the networks of both operators will be added onto the duration of the first full authentication of a roaming client. As Clancy et al. [BD09], we assume the added (possibly intercontinental) latency to be roughly 100-300ms [LGS07]. Communication originating in a FN and destined to the HN of the device will be affected by the increased latency. In particular bootstrapping IPsec between a roaming MC and the HN. Mechanisms relying on DSRKs exported from HN to FN, e.g., IPsec from MCs to MGs, or handovers in FN, do not suffer from increased latency, but will rather profit since communication with the HN is not necessary after the initial authentication.
6.6 Summary & Conclusion

In this chapter we proposed the first security architecture for WMNs that provides comprehensive multi-operator support in commercial as well as in community scenarios. Our multi-operator architecture extends our security architecture introduced in Chapter 5 and supports the secure deployment of all components in a WMN, i.e., MRs, MGs, and MCs. These components may be operated by a single operator or shared by different operators in order to achieve better network coverage and service quality. In addition, our architecture enables secure roaming of MCs between WMNs (with potentially mixed infrastructure). The deployment and management of network devices is based on EAP, RADIUS, IPsec, and a three party protocol allowing authenticated devices to bootstrap security associations with other devices, even when roaming to a FN. Compared to a single-operator architecture, our generalized proposal inflicts only a minimal performance penalty, which is due to the latency between networks of different operators across the Internet.

For practical purposes it is required to set up a key transport protocol between AAA domains. Today, the only keys that are transported form a device’s HN to the NAS of the FN are the PMKs used during the 4-way handshake. As such, the AAA server must support additional protocols, storage, and management tools to fully support a multi-operator architecture as proposed in this chapter.
CHAPTER 7

Handover Extensions

Mesh Clients (MCs) in a Wireless Mesh Network (WMN) are typically mobile devices and as such can move from the coverage area of one Mesh Router (MR) to the next during an ongoing connection such as a Voice over IP (VoIP) call. A handover procedure ensures that the MC can move from its Current Mesh Router (CMR) to the Target Mesh Router (TMR) without any disruption of its ongoing connections. The security challenge of a handover procedure is to ensure that the connection between MC and TMR can be adequately secured while keeping the delay constraints. In particular, the keying material required to protect the connection has to be efficiently established. Running a full authentication (e.g., Extensible Authentication Protocol (EAP)) via the TMR during handover for this purpose is typically too time consuming.

While there is no prior work on handover security targeted for WMNs, this topic has been extensively studied in infrastructure Wireless LAN (WLAN) [Cla08, CNND08, HNO10, MLOPG10]. However, these approaches share a major shortcoming that renders them hard to deploy to WMNs. They typically transfer keying material over a wired backbone to the TMR. In a WMN where all connections are wireless such an approach would obviously leak the transferred keying material easily, unless the connection between CMR and TMR is protected. The approaches above assume that keys can be distributed securely.

Two approaches are possible for handover, a proactive and a reactive approach. In a proactive handover procedure the keying material is allocated to a TMR prior to the actual handover procedure. In a reactive handover procedure the keying material is established right after the handover.

In this chapter, we propose three complementary secure, efficient and practical handover protocols for WMNs. The EAP-TTLS Neighborhood Pre-Authentication (ENPA), allows to proactively supply candidate TMRs with keying material as part of the initial EAP authentication of an MC joining the WMN. The other two protocols 3-Party Handshake for Handover (3PHSH) and Neighborhood Pre-Authentication (NPA) can be run at any point in time after successful authentication of the MC. The second of these protocols allows several TMRs to be proactively supplied with key-
ing material but the MC cannot be sure that these TMRs have already received the keying material at the end of the protocol run. The second protocol has the advantage that an MC initiating the protocol can be sure that the one TMR supplied with keying material has received it when the protocol terminates. We implemented and evaluated the three newly proposed protocols in our live WMN testbed (cf. Chapter 4) and integrated them into the de-facto standard WLAN software hostapd and wpa_supplicant. The results of our extensive performance evaluation are presented as part of this chapter.

Our handover protocols also easily function in a multi-operator WMN as described in Chapter 6. The only requirement is that involved devices, i.e., MC, CMR, TMR and the Authentication, Authorization and Accounting (AAA) server, have a security association based on the respective Domain Specific Root Key (DSRK). The details on handover in multi-operator environments are explained in Section 7.3.5.

The work presented in this chapter has also been published in [EHM13].

Outline: In Section 7.1 we briefly discuss related work on handover security in infrastructure WLAN and WMNs. Section 7.2 introduces handover related preliminaries. Next, Section 7.3 introduces our handover protocols, while their performance is evaluated in Section 7.4. After analyzing their security in Section 7.5, we conclude this chapter with a discussion in Section 7.3.4.

7.1 Related Work

In this section we briefly discuss the most prominent handover mechanisms for infrastructure WLAN.

Configuration And Provisioning for Wireless Access Points (CAPWAP) [OCK05] was designed to simplify deployment and management of enterprise WLAN infrastructures. The functionality of an Access Point (AP) is split into two components, i.e., Wireless Termination Point (WTP) and the Access Controller (AC). A WTP implements the Physical Layer (PHY) layer and lower portions of the Medium Access Control (MAC) layer functionalities. Note, that this includes that a Station (STA) is able to secure the communication with a WTP on the link layer using standard IEEE 802.11i [IEE04a] mechanisms. The second component is the AC which implements the upper portions of the MAC layer, including authentication and access control features. When a STA moves to another WTP it executes the 4-way handshake [IEE04a] with the AC instead of a full EAP authentication. Subsequently, the required keys are sent from the AC to the WTP reducing the communication between the STA and the AAA server which is typically further away. Therefore,
the ability to perform fast handover in CAPWAP is a side effect of moving the authentication process to a central component nearby. However, splitting the AP functionality makes it hard to deploy to standardized devices. Additionally, the secure key transport between WTPs, ACs and AAA server is not considered for WMNs where untrusted intermediaries may be present. In CAPWAP several ACs may exist which control a set of WTPs each. If a STA moves to a WTP belonging to different ACs, a full EAP authentication is still required. Therefore, CAPWAP does not necessarily prevent long handover delays.

The EAP re-Authentication Protocol (ERP) [ZHZ12] is a proposed standard of the IETF. Its purpose is to avoid a full EAP authentication when a STA re-authenticates. Depending on the EAP-method, multiple round-trips between the AAA server and the peer may be required. For example, the standard EAP-MD5 [BV98] method requires at least two round/trips. ERP supports intra- and inter/domain handover. ERP aims to reduce the handover disconnection time when a STA roams to another domain and after its first authentication in that domain. However, if deployed directly in a WMN, the transport of the re-authentication keys is not sufficiently protected against intermediaries as it is sent in a Remote Dial-in User Service (RADIUS) message (cf. Section 2.2) which is only protected by MD5 [RRSW97]. Therefore, it is possible for intermediate, untrustworthy nodes in a WMN to compromise the handover keys.

The IEEE 802.11r standard [IEE08] delivers a key to the AP to which the STA has first connected to. On handover this AP derives further keys and distributes them to the target AP to which the client is moving to. This approach cannot be used in WMNs since each APs may easily be compromised. In addition, the standard assumes pairwise keys between the APs to securely transport the handover keys without specifying how these keys are to be established and distributed.

In [MHSA04] Mi-Ho et al. propose a proactive key distribution to reduce the re-authentication delay. Keying material is distributed to neighboring APs of the serving AP. It is assumed that if the STA moves to another AP, it is likely to be one of the neighboring APs of the serving AP. Note that the AAA server is responsible for deriving and delivering the keys to the handover destinations. In a WMN the path from the AAA server to the TMR may contain untrustworthy nodes such that again this approach cannot be applied to WMNs directly. Also, the authors do not specify how the AAA server gains knowledge of the movement of the STA from one AP to another.

The above shows that most handover protocols proposed for infrastructure WLAN are of reactive nature and are thus only able to achieve a hard handover which tends to break ongoing sessions. In addition, secure key transport to the
handover destinations is either not addressed, or not completely solved in prior approaches. We address these shortcomings by introducing secure, efficient and practical proactive handover protocols achieving a soft handover.

7.2 Preliminaries

This section discusses handover related preliminaries that are important in this context. We start by introducing the network assumptions and the key distribution model. Additionally, we detail the IEEE 802.11 mechanisms that we leverage in order to create our handover protocols.

7.2.1 Network Assumptions

We assume the network to be operated by a single operator. WMN devices have been deployed according to Framework for establishing Security Associations for Sequentially Deployed WMN (FSASD), thus sharing the key hierarchy with the AAA server (cf. Chapter 5). As a result, the AAA server shares an Internet Protocol Security (IPsec) connection with each MR in the network. This represents a confidential, integrity protected, and authentic channel. As also the MC shares the key hierarchy with the AAA server, a secure channel can be established.

7.2.2 Key Distribution

Our proposed handover protocols assume one or more key distribution components to be present which are responsible for generating the handover key and distribute them to the requested TMRs of the MC. In coherence with the EAP security model, all handover keys are generated by the AAA server and the MC. In particular, the Extended Master Session Key (EMSK) never leaves the AAA server. Confidential, integrity protected, and authentic key transport from AAA to the TMRs is ensured using IPsec in all our proposed protocols. The required security association between the AAA server and the TMR is established when the TMR joins the WMN according to the FSASD mechanisms as detailed in Chapter 5. Note that transporting key material from a AAA server to Network Access Server (NAS) is covered by the standard use of EAP over RADIUS. However, pro-actively delivering handover keys to TMRs is not yet addressed by any standard.

We therefore designed and implemented a key transport protocol between the AAA server and the TMRs, which is used in our new protocols. In this protocol, the TMRs listen for incoming key deliveries from the AAA server. The IPsec Security Association (SA) which was created upon the initial authentication of the
TMR ensures that only authenticated and integrity protected key deliveries can be received on this connection. Confidential, integrity protected and authentic delivery of the parameters necessary to generate the handover keys at the MC is ensured by using the Root Peer Authentication Key (rPAK) of the FSASD key hierarchy.

As the distance between the key distribution component, the TMRs, and the MC is essential for the performance of the handover protocols, we propose to use the FSASD Authenticator (FAU) (cf. Section 5.3). The FAU is responsible for generating and distributing the handover keys to the respective TMRs, just as a AAA server would. However, as a component it can either be co-located with the AAA server, or be deployed to multiple MRs of the WMN, e.g., closer to the MCs. If it is co-located with the AAA server, secure communication between both components is trivial as it is the same physical entity. Otherwise, secure communication is ensured by an IPsec SA between the FAU (i.e., the MR it is running on) and the AAA server which has been bootstrapped during the MR’s authentication (cf. Chapter 5).

Based on the concept of Usage Specific Root Keys (USRKs) [SDNN08] only a subset instead of the full FSASD key hierarchy is exported by the AAA server to the FAUs. USRKs are keys that are bound to a specific use, e.g., for the sole purpose of generating handover keys for MCs. Thus, FAUs could be placed near the networks’ edge, i.e., close to MCs requiring handover services.

### 7.2.3 IEEE 802.11i PMKSA-Cache

The Pairwise Master Key Security Association (PMKSA) describes a security association between STA and AP. The Pairwise Master Key (PMK) is used between STA and AP to carry out the IEEE 802.11i 4-way handshake (cf. Figure 2.5) which generates keys that are subsequently used to secure the link layer traffic between both parties. The PMK is stored with context information such as the AP MAC address, the lifetime of the PMK, and a unique ID called Pairwise Master Key ID (PMKID) (cf. Equation 7.2). If a STA and AP share a PMKSA, e.g., because the STA was connected to the AP before, both can use the cached PMK in the 4-way handshake directly [IEE04a] instead of running EAP to establish a fresh PMK. For this purpose the STA retrieves the MAC address of the AP from its beacons and sends a (Re-)Association Request to the STA including the PMKID in the Robust Security Network (RSN) Information Element (IE) of the request. If the AP determines that it has cached the respective PMK for the PMKID, it directly starts the 4-way handshake with the STA. Otherwise, the STA has to run a full EAP authentication which takes significantly longer.
7.2.4 Vendor Specific Information Element

For implementation purposes we extended the IEEE 802.11 wireless beacon using a vendor specific IE containing the Internet Protocol (IP) address of the TMR. The MC can obtain the IP address from the beacon to initiate communication with the TMR on the network layer. If our protocol would be run in a network environment in which multi-hop communication on the MAC layer is possible, advertising the TMR’s IP in the beacons would not be necessary. Instead, the wireless frame transporting the protocol messages would simply be forwarded by the intermediate hops to the destination.

Section 2.3 described the IEEE 802.11 wireless beacons and its vendor specific IE. We use the beacon to define and transport our own IE containing the IP address of the MR broadcasting its beacon. By retrieving the IPs from the beacon, the MCs is able to obtain information how to address the MRs on the network layer. In order to transport the IP address we specify an Organizationally Unique Identifier (OUI) and include the IP as the content. Note, that this procedure is in compliance with the IEEE 802.11 standard.

As Section 2.3 has also shown, the standardization efforts of the IEEE have realized that some services, e.g., handover, STAs in a pre-association state, can profit by obtaining meta-information of the surrounding stations.

7.3 Handover Protocols

In this section we present our three novel proactive handover protocols for WMNs with different features, namely the 3PHSH, NPA, and ENPA. We build onto the EAP-based network deployment strategy introduced in Chapter 5 and the respective security associations (e.g., IPsec between MRs and the AAA server).

The goal of the protocols is to establish PMKs as known from IEEE 802.11i (cf. Section 2.3) between the MC and the handover destination TMRs. Once the MC decides to associate with one of the available TMRs, both can simply use the established PMK to carry out the 4-way handshake instead of running a full EAP authentication. Thus, the re-association delay (or handover delay) when moving from one MR to another can be greatly reduced. Even though our protocols operate in an EAP-based network deployment, topology management by the AAA server is not required.
7.3. Handover Protocols

7.3.1 3-Party Handshake Protocol for Handover (3PHSH)

3PHSH is an extension of the three party protocol proposed in Section 5.2.5. In the context of handover, we adapted the original proposal such that the three parties are an MC, an MR as the handover target, and the AAA server which is involved in the handover key generation (cf. Figure 7.2). We assume that both, the MC and the TMR have been authenticated using EAP according to FSASD (cf. Chapter 5), i.e., they share a specific set of cryptographic keys with the AAA server which are essential to securing the key derivation and the key transport.

An MC can initiate the protocol with a specific TMR. 3PHSH consists of the messages shown in Figure 7.1. Message M1 contains the PAKID which is used by the AAA server to map a rPAK to a specific MC, as it needs to be able to decrypt parts of the message. The content is encrypted using the rPAK shared between the MC and the AAA server. Once the TMR receives M1, it appends a nonce (N_{TMR}) to the message and relays it to the AAA server via a secure channel, i.e., the IPsec
connection established between the TMR and the AAA server, as message M2. After receiving M2, the AAA server generates the PMK to be shared between the MC and the TMR (cf. Equation 7.1).

\[
PMK_{TMR} = \text{KDF}(rKDK_{MC,AAA}||N_{TMR}||N_{AAA} || N_{MC}||MAC_{MC}||MAC_{TMR})
\] (7.1)

The key derivation function (cf. Equation 7.1) is keyed with the rKDK_{MC,AAA} shared between the MC and the AAA server according to the FSASD key hierarchy (cf. Figure 5.2). Additionally, a key label is required, as was well as random nonces of the three parties, and the MAC addresses of the MC and the TMR.

Now the AAA server delivers the encrypted contents needed by the MC for generating the PMK to the TMR as Message M3. It also appends the PMK and the MAC address of the MC. The MAC address is used by the TMR as an input to generate the PMKID (cf. Equation 7.2) which is used to identify the MC and the PMK to be used during the 4-way handshake. Finally, the TMR only forwards the encrypted parameters to the MC as Message M4. Now the MC is able to generate and insert the PMK into its PMKSA-cache [IEE04a]. Based on the PMK and the corresponding PMKID, the MC can now initiate the 4-way handshake with the TMR.

\[
\text{PMKID} = \text{HMAC-SHA1-128}(\text{PMK}, \text{key_label}||\text{MAC}_{TMR}||\text{MAC}_{MC})
\] (7.2)
7.3.2 Neighborhood Pre-Authentication (NPA)

The so-called Neighborhood Pre-Authentication (NPA) protocol is similar to 3PHSH. It can be triggered anytime by an MC after its authentication. As opposed to 3PHSH, NPA is able to initialize multiple handover TMRs in a single protocol run. Additionally, the message overhead is reduced by two messages due to directly communicating with the AAA server instead of using the TMR as an intermediary.

Figure 7.3 depicts a protocol run of NPA. In this example the MC’s goal is to establish PMKs with TMR₁ and TMRₙ which can potentially be used for handover.

An NPA protocol run consists of the messages shown in Figure 7.4. In Message M₁ the MC requests PMKs to be generated by the AAA server for a set of potential handover TMRs. It includes the relevant parameters to generate the PMKs, i.e., the MC’s MAC address, a random nonce and a timestamp of the MC, and most importantly, the list of IP addresses identifying the TMRs. Those are necessary as the AAA server needs to be able to address the TMRs for the purpose of delivering the handover keys. The mentioned parameters are encrypted using the rPAKₜₕₜₜ which is shared with the AAA server. The PAKID is used to map the MC’s identity at the AAA server.

Once the MC has received Message M₂ from the AAA server, it can derive the PMKs using the provided inputs, as well as the IP address of the respective TMR (cf. Equation 7.3). Again, as in 3PHSH, the IP addresses of possible handover TMRs are obtained using the vendor specific IE of the IEEE 802.11 beacon.

\[
\text{PMK}_{\text{TMR}_i} = \text{KDF}(r\text{KDK}, \text{key\_label}||\text{MAC}_{\text{MC}}||\text{N}_\text{MC}||\text{N}_\text{AAA}||\text{IP}_{\text{TMR}_1}) \tag{7.3}
\]

The AAA server also sends a message containing the MC’s MAC address and the individually generated PMK to each TMR requested by the MC. Each TMR can now generate the PMKID (cf. Equation 7.2) used to map the PMK and insert
it into its PMKSA-cache. The MC is now able to use the established PMK during handover to an initialized TMR by sending the respective PMKID in an association request. If a mapping is found both can directly start the 4-way handshake. The MC determines the MAC address of the TMR from its 802.11 beacon.

7.3.3 EAP-TTLS Neighborhood Pre-Authentication (ENPA)

3PHSH (cf. Section 7.3.1) and NPA (cf. Section 7.3.2) are suitable to be run after an initial authentication of the MC. Thus, we propose another mechanism to initialize multiple TMRs for handover directly during the initial authentication. Additionally, ENPA can be run automatically whenever a full EAP authentication becomes necessary, e.g., when the EAP session times out, or the device has connected to another MR without using a handover mechanism. Figure 7.5 shows the general idea of ENPA.

ENPA is currently realized as an extension of the EAP-TTLS [PF08] authentication method. However, it can easily be applied to any other EAP method that allows the transport of Diameter Attribute Value Pairs (AVPs) [PJE+03]. The AVPs used by EAP-TTLS and Diameter are syntactically equivalent. Figure 7.7 shows the AVP format used by ENPA. As in the protocols described in the previous sections, the MC itself is responsible for specifying for which TMRs handover keys should be established.

When an MC associates with the network it scans its surrounding and acquires a number of available TMRs. It also retrieves their IP addresses from the IEEE 802.11 beacon. The MC embeds the IP addresses of the TMRs it chose to prepare for a potential handover along with its own MAC address. The MC’s MAC address must
be used, as it is required by TMRs to generate the PMKID (cf. Equation 7.2) to map the handover PMK to an associating MC. See Figure 7.6 for the message flow of ENPA.

\[
PMK_{TMR_i} = KDF(rKDK, key\_label||IP_{TMR_i}||
\text{tls\_client\_random}||\text{tls\_server\_random})
\]  

(7.4)

The AAA server generates distinct PMKs for each of the embedded IP addresses received from the MC in the AVP. The key derivation uses the rKDK of the FSASD key hierarchy, a key label, and the individual IP address of the respective TMR (cf. Equation 7.4). Additionally, tls_client_random and tls_server_random of the EAP-TTLS session are used as salt values similar to the nonces in 3PHSH and NPA. We use these values as they have been exchanged during the Transport Layer Security (TLS) handshake of the EAP method anyway. Also, it allows us to bind the key generation to a specific TLS session.
Once the AAA server has generated the PMKs it sends the key, along with the MC’s MAC address to the TMRs. Note that this particular key transport is encrypted and integrity protected by an IPsec security association between the AAA server and the TMR according to FSASD (cf. Chapter 5).

The AAA server sends the necessary key derivation parameters to the MC in an AVP of the RADIUS-Access-Accept message which marks the end of a successful EAP authentication. The MC can now derive the PMKs, generate PMKIDs and insert them into its PMKSA-cache.

Once a handover becomes necessary, the MC simply selects a corresponding PMK and queries the TMR with the according PMKID used in the association request. MC and TMR can then carry out the 4-way handshake based on the PMK.

### 7.3.4 Discussion

The proposed protocols 3PHSH, NPA, and ENPA are each proactive and can be used at different epochs of a MC’s network connection. Each is able to proactively, i.e., before it is actually necessary, establish fresh PMKs with TMRs that enable the MC a fast and efficient association and authentication based on the 802.11 4-way handshake. ENPA is used at the very beginning of a connection, and can be used anytime the EAP session is refreshed. 3PHSH and NPA are both post-authentication protocols, i.e., they are to be used after an initial network association and EAP-authentication. NPA being more efficient in terms of the communication overhead allows to prepare multiple TMRs for fast handover of MCs, whereas each 3PHSH protocol run bootstraps a single handover destination while offering TMR consent.

The message overhead of 3PHSH is \(4 \times n\), and \(2 + n\) respectively for NPA where \(n\) is the number of TMRs for which PMKs are requested. Using ENPA is not considered to be time critical, as MCs will associate with the CMR after EAP authentication rather than directly handover. Its message overhead is \(n + m\) where \(m\) is the number of messages of the EAP method. Using EAP-TTLS results in \(m \geq 4\). Thus, the protocols can be used alongside each other; ENPA whenever a full EAP-authentication becomes necessary, and 3PHSH or NPA as the MC moves through the WMN.
### 7.3.5 Handover in Multi-Operator Scenarios

In Chapter 6 we described how our comprehensive security architecture (presented in Chapter 5) can be used in multi-operator scenarios. The handover protocols proposed in this chapter are obviously also relevant to these scenarios. We will now elaborate on this issue assuming a MC\(\pi\)\(\psi\) that is currently roaming in the domain \(\psi\) using the same notation as in Chapter 6.

Section 6.3 introduced the concept of so-called DSRKs which enable services in specified domains. These services may also resemble handover. As such, the protocols and keys generated in this chapter will differ only in the selection of keys that are used.

### 3PHSH

This protocol can be used in a FN, e.g., domain \(\psi\), as the local AAA\(\psi\) will be in possession of the DSRK\(\psi\) of a roaming device MC\(\pi\)\(\psi\). Figure 7.9 depicts a snippet of the domain specific key hierarchy introduced in Section 6.3. To achieve message authentication and confidentiality to AAA\(\psi\), the MC\(\pi\)\(\psi\) now uses rPAK\(\psi\) which is valid to be used in the domain \(\psi\). The CMR\(\psi\) (regardless of operator allegiance - cf. Section 6.3.3) relays this message via a secure IPsec connection to the AAA\(\psi\). It is secured by the domain specific keys of the CMR\(\psi\), TIK\(\psi\) and TEK\(\psi\).

MC\(\psi\) and TMR\(\psi\) agree upon PMK\(\psi\)\(\text{TMR}_i\) which is generated using the key derivation show in Equation 7.1 keyed with rKDK\(\psi\) of MC\(\psi\).

3PHSH can also leverage the concept of the FAU (cf. Section 5.3) while roaming in a FN. Now, instead of using the rPAK\(\psi\) with the local AAA server, the node uses the PAK\(\psi\)\(\text{FD}_i\) with the responsible FAU\(\psi\). As such, also the key used during the PMK derivation will differ and use the PMKDK\(\psi\)\(\text{FD}_i\) (cf. Figure 7.9).
Figure 7.9: A snippet of the domain specific FSASD key hierarchy including keys used by the logical FAU domains.

NPA

NPA will be equally useful in a multi-operator setting as 3PHSH. The necessary domains specific keys are exchanged in the same fashion as in the previous paragraph. That is, MC$^\pi$ either uses rPAK$^\psi$ or PAK$^\psi_{FD_i}$ to secure its message with either the local AAA server or a FAU of the current roaming domain $\psi$. The derivation of PMK is either based on the rKDK$^\psi$ if the AAA server is involved, or based on the PMKDK$^\psi_{FD_i}$ if done by a FAU.

ENPA

As the two previous protocols, ENPA can easily be used in a multi-operator scenario. This authentication of the roaming MC$^\pi$ using the home AAA$^\pi$ is carried out as described in Section 6.3.2. Recall, that the AAA$^\pi$ sends the DSRK$^\psi$ of the MC to AAA$^\psi$ after its authentication. In order to generate the PMKs the rKDK$^\psi$ is used by the AAA$^\psi$ in a modified key derivation (cf. Equation 7.5) without the random values of the TLS session. The vanilla ENPA uses these values to increase the length of the salt used during the key derivation.

$$PMK_{TMR_i} = \text{KDF}(rKDK^\psi, \text{key_label}|\text{IP}_{TMR_i}) \quad (7.5)$$

Note that in the domain $\psi$ each infrastructure device bootstraps a security
association to the local AAA server, i.e., $\text{AAA}^\psi$ (cf. Section 6.3.3). As such, $\text{AAA}^\psi$ can send the generated PMKs to the TMRs requested by $\text{MC}^\psi$.

In order to prevent intermediate nodes from tracking possible handover destinations of MCs, the IPs of the TMRs are encapsulated in the TLS session. As only the home AAA server can extract them, it is necessary to relay these IPs back to the foreign AAA server as they are needed for the key derivation.

Similarly, the vanilla-ENPA, the multi-operator version cannot profit from the FAUs deployed in the WMN.

7.4 Performance Evaluation

This section presents the performance evaluation of our proactive handover protocols $3\text{PHSH}$, NPA and ENPA using our live WMN testbed. The measurements compare the protocols running on top of the two most popular transport protocols TCP and UDP. Additionally, implicitly relevant factors for handover, i.e., the IEEE 802.11i 4-way handshake and the wireless scanning process are measured.

7.4.1 Theoretical Handover Performance

As practical evaluation and comparison of the related handover mechanisms for infrastructure WLAN (cf. Section 7.1) is not possible, due to either specialized hardware that is unavailable and missing implementations, we compare the protocols on an analytical level.

We use the analytical method and definitions proposed in [Cla08]. Here we specifically analyze the runtime of handover-specific protocols. This does not include the 4-way handshake as it is not specific to a handover protocol but needs to be run in a regular wireless association of STA and AP.

Let

- $N_e$ be the number of round trips for a specific EAP method;
- $T_w$ the transmission latency between STA and AP;
- $T_c$ the latency between any two relatively close devices;
- $T_a$ be the latency between infrastructure components and the AAA server;
- $T_{hs} = 4 \cdot T_w$ be the duration of the 802.11i 4-way handshake.

Thus, the initial IEEE 802.11i transition, which actually corresponds to a full EAP authentication, requires $2 \cdot N_e \cdot (T_w + T_a) + T_a$. For CAPWAP one can obtain
4 \cdot (T_w + T_c) + T_c, for HOKEY 2 \cdot (T_w + T_a), and 2 \cdot T_w + 2 \cdot T_c + 2 \cdot T_w for the IEEE 802.11r [Cla08].

Assuming the round trips and latencies of our testbed (cf. Chapter 4) as: \( N_e = 4, T_w = 1.86 \text{ ms}, T_c = 3.57 \text{ ms}, \) and a four hop distance to the AAA server with \( T_a = 7.02 \text{ ms}, \) yields the estimated handover performance depicted in Table 7.2.

As we will see in the following, the individual runtimes of our three handover protocols can match the theoretical performance computed analytically. Thus, we can show that our protocols are competitive to the other approaches even though they operate on a wireless instead of a wired backbone.

The analytical runtime of our protocols are:

- **3PHSH**: \( 2 \cdot (T_w + T_c) + 2 \cdot T_a \), MC-AAA round trip; latencies MC-CMR/TMR
- **NPA**: \( 2 \cdot T_a + 1 \cdot T_a \), MC-AAA round trip; unicast to one AAA-TMR
- **ENPA**: \( 2 \cdot N_e \cdot (T_w + T_a) + T_a \), as IEEE 802.11i, using one TMR.

Also note that as opposed to our handover protocols, all related approaches compared in this section are only applicable to a single handover destination. But both NPA and ENPA are capable to prepare multiple handover destinations at once while only requiring an additional unicast from the AAA server to the TMR. For the purpose of a fair comparison, we assumed both to be run for a single TMR.

The runtime of NPA for multiple TMRs is in the worst case bound by the key delivery time to either a requested TMR, or to the MC itself (cf. Section 7.4.4). A handover can only be successfully be initiated if both the TMR and the MC are in possession of the handover key. The same holds true for ENPA which delivers the key to the MC inside of the EAP method and uses a single unicast message to deliver the keys to the TMRs.

This means that the theoretical protocol runtime (cf. Table 7.2) for bootstrapping one TMR cannot be used directly to estimate the runtime for bootstrapping multiple TMRs in a scalar manner.

### 7.4.2 Methodology

After discussing the handover performance of our protocols on a theoretical level, we now continue with the methodology that we used to evaluate the protocols in our real world testbed.

All measurements of 3PHSH, NPA, ENPA, the 4-way handshake, and the IEEE 802.11 scanning process have been repeated 100 times in each run to obtain sufficiently stable results. The results depicted in the figures represent median values,
### Table 7.2

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Theoretical Runtime (ms) for one TMR</th>
<th>TMRs</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11i [IEE04a]</td>
<td>85.5</td>
<td>1</td>
</tr>
<tr>
<td>802.11r [IEE08]</td>
<td>14.85</td>
<td>1</td>
</tr>
<tr>
<td>HOKEY [Cla08]</td>
<td>25.2</td>
<td>1</td>
</tr>
<tr>
<td>CAPWAP [OCK05]</td>
<td>25.29</td>
<td>1</td>
</tr>
<tr>
<td>3PHSH</td>
<td>24.9</td>
<td>1</td>
</tr>
<tr>
<td>NPA</td>
<td>21.06</td>
<td>≥ 1</td>
</tr>
<tr>
<td>ENPA</td>
<td>85.5</td>
<td>≥ 1</td>
</tr>
</tbody>
</table>

Table 7.2: A comparison of the theoretical handover duration for bootstrapping a single handover destination. Note that only NPA and ENPA are capable of bootstrapping multiple handover destinations in one protocol run.

![Evaluation Topology](image)

**Figure 7.10:** The general purpose evaluation topology that is used for 3PHSH, NPA, and ENPA.

For each protocol we created specific network topologies to be able to hand-tune the relevant factors, such as the distance between the involved network components.

#### 7.4.3 3PHSH Evaluation

Figure 7.10 shows the topology that has been used to evaluate the performance of 3PHSH. The purpose of this specific network topology is to increase the number of wireless hops between the MC requesting handover and the AAA server generating and distributing the handover keys by one in each test run. In the i-th test run, the MC is connected to MR\(_i\) (CMR) which is i hops from the AAA server. Now the MC requests a handover key for MR\(_{(i+1)}\) (TMR).

During the evaluation, the distance from MC to the AAA server proved to be most relevant for the protocol’s runtime. Figure 7.11 shows the runtime of 3PHSH...
using TCP and UDP. In order to make a qualitative statement about the link quality of the communication path, we additionally included the latency between the MC and the TMR. UDP is roughly 50% faster (average) than TCP which can be accounted for by the absence of a handshaking mechanism. However, both are obviously influenced by the current link quality. The maximum message size of 3PHSH is 107 byte, which is far less than the Maximum Transmission Unit (MTU) of the testbed (1528 byte). Thus, this result supports the conclusion that the runtime of the protocol is directly proportional to the cumulative link latencies between MC and the AAA server, as well as between TMR and the AAA server.
Figure 7.13: The NPA star topology focussing on the amount of requested TMRs.

Figure 7.14: This figure shows the duration after NPA message M2 containing the PMK is received at the TMRs for various distances using TCP and UDP.

7.4.4 NPA Evaluation

NPA has been evaluated using the topologies shown in Figure 7.10, Figure 7.12, and a star topology in which the distance between all TMR and the MC is one (cf. Figure 7.13). Additionally, we evaluated how the number of TMRs included in the MC’s request influences the protocol runtime. Again, both UDP and TCP were compared as alternative transport protocols.

At first, the topology shown in Figure 7.12 was used. Each TMR was at least two hops from the AAA server. The CMR to which the MC is connected remains MR1 in each run. In our measurements we iteratively increased the maximum distance between the requested TMRs and the AAA server by one hop. Additionally, we increased the amount of requested TMRs by one. In the last run the MC requests PMKs for MR3, MR4, MR5, MR6, and MR7 which is four wireless hops from the AAA server.
To determine the relevant runtime we computed two values: $\Delta t_{MC}$ which is the time the MC receives the final message from the AAA server enabling it to derive the PMKs. And $\Delta t_{TMR_i}$ marks the time the TMRs receive the respective PMK from the AAA server. For the MC to make a successful handover it must possess the handover key, but also the TMRs must possess the handover key. In any cases, whichever key delivery path takes longer marks the end of a protocol run, as otherwise a handover will be unsuccessful.

Figure 7.14 shows the result of the last run in which PMKs for all TMRs are requested. The values on the x-axis denote the specific TMRs. The respective y-axis shows the time it took till M2 is received at the TMR. As expected, sequentially delivering the PMKs using TCP has a linear effect on the runtime. Delivering a total of five PMKs to MR3-MR7 takes approximately 50 ms using TCP, and 10 ms using UDP.

Figure 7.15 shows the duration until the PMKs are received at the TMRs for both TCP and UDP. Again, UDP is approximately proportional to the latency between the AAA server, TMR, and MC.

Additional measurements using a star topology (cf. Figure 7.13) revealed that the distance to the AAA server is the decisive factor rather than the number of requested TMRs. When using TCP to deliver the PMKs to the TMRs, its handshake plays into the overall runtime. Parallelization of PMK delivery is likely to reduce the runtime down to approximately the time it takes to deliver a PMK to a single TMR in the best case.

Using the topology shown in Figure 7.10 we varied the distance between the MC and the AAA server in the same way as for the evaluation of 3PHSH. We observed that the distance between MC and AAA server has very similar effects on
7.4. Performance Evaluation

The runtime (cf. Figure 7.16). The communication performance to the AAA server is proportional to the current latency on the path. Again UDP outperformed TCP.

7.4.5 ENPA Evaluation

By considering the analysis results of 3PHSH and NPA, and also prior results shown in Section 5.5, it becomes evident that the number of wireless hops (MC and TMRs) to the key distribution component, i.e., the AAA server, is the decisive factor for the protocols’ runtime. Thus, we used the topology shown in Figure 7.10 in order to evaluate both vanilla EAP-TTLS and our extension, ENPA. The runtime of vanilla EAP-TTLS can be seen in Figure 5.7.

In Figure 7.17 the results for ENPA are shown. Because the results for EAP-TTLS as shown in Figure 5.7 already revealed the effects of varying distance between the MC and the AAA server, we fixed the number of wireless hops between the MC and the AAA server to one. Instead we varied the number of IP addresses included in the EAP-TTLS AVP as this is the only difference in communication between the MC and the AAA server. Considering the variability of the wireless spectrum, we can conclude that increasing the payload \((\leq\text{MTU of the network})\) has an almost negligible effect on the runtime. This is due to the fact that ENPA uses the same mechanism (UDP) to deliver the PMKs to the TMRs.

The runtime for EAP-TTLS increases roughly about 20 ms per wireless hop, whereas adding an IPv4 address increases the duration about 7 ms. Considering that ENPA is envisioned to be used during initial authentication, and possibly when the EAP session times out, this increase in the protocol runtime is negligible. The other runtime component, i.e., the unicast message from the AAA server to
the TMRs, does not influence the runtime of the protocol negatively. It is roughly proportional to the latency between the AAA server and the TMR and it is thus very likely less than the runtime measured at the MC.

7.4.6 IEEE 802.11 Specifics

In addition to the performance of our protocols, we were also interested in the runtime of scanning the wireless spectrum for access points, and the duration of the 802.11i 4-way handshake. The scanning process is of practical relevance as the MC needs to discover relevant TMRs. Current wireless drivers and the wpa_supplicant implementation have slightly contradicting Basic Service Set Identifier (BSSID) caching policies such that it may happen that the driver forgot a BSSID, even though it should still be available. This forces current implementations to re-initiate a costly scan of the spectrum. However, this can be done periodically without interfering with the current connection. Figure 7.18 shows the duration of the scanning process in relation to the number of frequencies. Each frequency accounts for approximately 110 ms. On the other hand we also analyzed the runtime of the 4-way handshake, which resulted in an average runtime of approximately 3.62 ms over 100 runs. It thus has a negligible effect for the cumulative handover duration.

7.4.7 Evaluation Summary

This section presented an extensive performance evaluation of our three handover protocols. Additionally, we also evaluated IEEE 802.11 specific metrics such as the
duration of the 4-way handshake and the process of scanning for available BSS’es.

We have seen that the handover protocols 3PHSH, NPA, and ENPA are highly influenced by the distance of the involved nodes to the key distribution component, e.g., the AAA-server. In Section 5.3 we proposed a key distribution component (FAU) which can be placed somewhere on the path between the nodes and the AAA-server. Thus, by using the FAU, the distance to the key distribution component can be greatly reduced. This will positively impact the runtime of 3PHSH and NPA as the latency to the key distribution component is a decisive factor. The ENPA protocol cannot be optimized in the same manner as the FAU does not store any user credentials. However, these credentials are necessary for authentication during EAP-TTLS. Since ENPA is envisioned to be used either during initial authentication of nodes, or anytime the EAP session times out, the runtime optimization would be less effective anyway as the context is not as time critical.

In terms of the theoretical protocol runtime discussed in Section 7.4.1 we can conclude that these values can be met for 3PHSH and NPA. The values for ENPA deviate somewhat as the practical evaluation includes establishing a TCP connection (3-way handshake) and a TLS handshake between the MC and the TLS server of the AAA server.

Besides improving the runtime of the handover protocols, also the performance of the IEEE 802.11 hardware and driver play an important role. We expect that the hardware support and the scanning procedures could be optimized significantly in future versions of the drivers. Also, handing information from the driver to the wpa_supplicant in an efficient and reliable way would be a great improvement from which handover protocols can benefit.

**Figure 7.18**: This figure shows the duration of scanning each channel using IEEE 802.11a and IEEE 802.11b.
7.5 Security Considerations

As for the original FSASD security architecture introduced in Chapter 5, we chose to base our security analysis on RFC4962 [HA07] “Guidance for AAA Key Management” as we rely on a AAA infrastructure. All our proposed protocols exploit the security properties provided by an FSASD deployed WMN, especially the fact that each MR has an IPsec channel to the AAA server.

Table 7.3 shows an overview of the results of security analysis. Each of our protocols provides the same security properties required in RFC4962.

(1) **Cryptographic algorithm independence** is achieved by our proposals as they use specific instances of cryptographic algorithms which can easily be replaced by alternatives without affecting the protocols.

(2) Each protocol execution produces **strong fresh session keys**, i.e., the PMKs of 256 bits in length. The keys are generated using PRF+ which is recommended and the current best practice [Kau10].

(3) The keys used to generate PMKs, as well as the handover keys themselves have a **clearly defined scope**. rPAKs and rKDKs shared between a node and the AAA are used to secure the protocol messages and generate handover keys. The PMKs on the other hand are only used during the 4-way handshake.

(4) **Replay protection** related to PMK delivery from AAA to the TMRs is achieved by their mutual IPsec SA. Protocol messages of 3PHSH and NPA between the MC and AAA rely on time stamps to detect message replay. Loose time synchronization is required.

(5) **All parties are authenticated** during network deployment by the AAA based on their individual credentials. Message authenticity is ensured by IPsec between AAA and the TMRs, and rPAK between MCs, AAA, and TMRs.

(6) The involved parties **implicitly demonstrated possession** of relevant keys in each protocol. If a party does not possess the respective key, it is unable to successfully send and receive messages that will be processed by the other parties. This is either enforced by IPsec, or by MACs on the message content.

(7) **Confidentiality of keying material** is either ensured by IPsec between the AAA server and the TRMs, or by encryption based on the rPAK. Keying material transported from and to MCs is secured by the rPAK.

(8) As our protocols neither support **ciphersuite negotiation**, nor do they support cipher suite confirmation.
7.6 Summary & Conclusion

In this chapter we proposed novel proactive handover protocols for WMNs which are secure, efficient and practical. Table 7.1 summarizes their key properties. Contrasting to the highlighted proposals for infrastructure WLAN, our solutions do not suffer from a bootstrapping problem. Using our protocols alongside the FSASD architecture enables us to meet a comprehensive set of security requirements for protocols in the context of AAA key management. 3PHSH, NPA, and ENPA are envisioned to be used in an interplay allowing to proactively instantiate handover candidates as the MC strides through different epochs of its network session. The practical evaluation using a live WMN testbed allowed us to determine the performance of the protocols, and additionally profile the related wireless properties.

### Table 7.3: This table summarizes the security analysis respective to RFC4962.

<table>
<thead>
<tr>
<th>Requirement (1-11)</th>
<th>3PHSH</th>
<th>NPA</th>
<th>ENPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Crypto-algorithm independent</td>
<td>yes (no negotiation)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Strong fresh session keys</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>3. Limit key scope</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>4. Replay detection mechanism</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>5. Authenticate all parties</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>6. Peer and authenticator authorization</td>
<td>indirect</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Keying material and confidentiality and integrity</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>8. Confirm cipher suite selection</td>
<td>no (no negotiation)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Uniquely names keys</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>10. Prevent the Domino effect</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>11. Bind key to its context</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

(9) All keys are **uniquely named** by using a key label which is strongly related to key usage.

(10) Authenticators, i.e., CMRs and TMRs only hold a limited amount of key material with a specific lifetime. Compromise only allows to access current and new keying material associated with this specific authenticator; others are not directly affected. Thus, the **domino effect** is prevented.

(11) **Key context** is explicitly established during key generation using the key label (e.g., cf. Equation 7.1).
of the 4-way handshake and scanning the spectrum. Altogether, the resulting performance highlights the applicability of our protocols in a time critical context, without negatively impacting ongoing sessions. Correctly deciding the point in time a handover would be beneficial, as well as designing a service to announce and discover handover candidates or key distribution components near to MCs leaves room for further research.
Part IV

Synopsis
In this thesis we proposed a comprehensive security architecture for Wireless Mesh Networks (WMNs) which copes with multi-operator scenarios and enables handover services for end-users as well as for the network infrastructure.

Prior to introducing the architecture we thoroughly revisited the technical foundations (cf. Chapter 2) we make use of, such as the IEEE 802.11i security model, the Remote Dial-in User Service (RADIUS) protocol, the Internet Protocol Security (IPsec) protocol suite, a bi-directional secure Remote Procedure Call (RPC) framework, and the Extensible Authentication Protocol (EAP) protocol.

After describing the necessary technical fundamentals, this thesis continued by introducing WMNs on a general level to establish the baseline needed to derive their security requirements (cf. Chapter 3). We specifically categorized the nodes in operator-centric WMNs to be Mesh Clients (MCs), Mesh Routers (MRs), Mesh Gateways (MGs), and Authentication, Authorization and Accounting (AAA) servers. Each of them has a clearly defined purpose in the network. Considering the node types and the resulting network topologies allowed us to define all the important communication patterns in WMNs such as: MC ↔ MC, MC ↔ MR, MC ↔ MG, MR ↔ MR, MR ↔ MG, and any communication involving the AAA server. To the best of our knowledge we were the first to propose such an approach when designing a security architecture for WMNs. The related work in this area usually focused on security of a single specific communication pattern, rendering the other patterns non-secured. Thus, a comprehensive and general approach did not exist so far.

We continued by defining an attacker model and studied the specific challenges WMNs face. These include different operator models (single- and multi-operator) where operators may even cooperate and share parts of their network infrastructure to form a single network. We also discussed the possible heterogeneity for the wireless links between devices, the challenges introduced by routing MCs, and mobility in general, i.e., that of MCs, but also mobility of parts of the network infrastructure. After that we used these assumptions to derive a set of general security requirements for WMNs and evaluated them against related research that aims at creating security architectures for WMNs. This study reveals a selection of gaps that our comprehensive security should address.
In Chapter 4 we presented the ITsec Testbed which has been designed in the process of developing our security architecture. Its main goal was to demonstrate the feasibility of our mechanisms and to provide a platform to evaluate the performance of our security protocols on real hardware instead of relying on simulations. Our testbed is based on Off-the-Shelf (OTS) hardware and comprised of about 20 nodes running standard software such as a Linux operating system and its respective wireless tools wpa_supplicant and hostapd. The testbed is modeled to represent a WMN managed by an operator. For this purpose we introduced a central management server running a AAA server. We have used the testbed to implement and evaluate our security architecture including the handover services (cf. Chapter 7).

Following the foundations lain out in the previous chapters Chapter 5 continues by proposing the central element of this thesis. The Framework for establishing Security Associations for Sequentially Deployed WMN (FSASD) is a comprehensive security architecture for WMNs which addresses all the security requirements derived in Chapter 3. We build on standard mechanisms such as EAP, IPsec, and the AAA protocol RADIUS. We use the EAP protocol and the Extended Master Session Key (EMSK) as a basis to construct an extensible key hierarchy that is used to bootstrap security associations between devices in the WMN. In FSASD each device is authenticated using EAP, thus, each authenticated device shares the key hierarchy with the AAA server. After connecting to the network, devices automatically bootstrap an IPsec Security Association (SA) with the AAA server. This is necessary to secure RADIUS traffic that is generated once another devices connects via the device. Also, MCs use the key hierarchy to setup an IPsec connection to MGs in the network to secure their multi-hop traffic. For this purpose we introduced the 3-Party Handshake for Sequential Deployment (3PHSD) protocol. This protocol allows an authenticated key agreement between two nodes that do not yet share a secret. We leverage the fact that each node shares a personal instance of the key hierarchy with the AAA server. As the AAA server is the main key distribution component, our security architecture also proposes an optimization by introducing the FSASD Authenticator (FAU). Not only does the FAU allow to segment the network into logical key distribution domains, but it also allows to speed up the 3PHSD protocol, as well a the handover protocols proposed in Chapter 7.

As the general design of our comprehensive security architecture implicitly supports an operator-based approach to WMNs, Chapter 6 generalizes it such that it enables multi-operator support and infrastructure sharing. In less complicated scenarios, such as roaming, MCs can use other networks maintained by different Foreign Operators (FOs) while being enrolled with their Home Operator (HO). By using the concept Domain Specific Root Keys (DSRKS) all the security features of
FSASD remain usable as a sub-level of the original FSASD key hierarchy is spawned for specific networks. This for instance allows roaming MCs to setup an IPsec tunnel connection to either a gateway of its HO (similar to a Virtual Private Network (VPN)), or to use the same mechanism with a gateway of the FO but with domain specific keys. In more complex scenarios the network is composed of devices from multiple operator to form a mixed network. Here, devices are connected to each other irrelevant of their individual operator allegiances. Still, based on the FSASD key hierarchy generalization our approach holds up to the security requirements proposed in Chapter 3.

While Chapter 5 and Chapter 6 laid out the architectural details, Chapter 7 presents three secure handover protocols as a significant application of the FSASD security architecture. The handover protocols interface with the extensible key hierarchy in single- and multi-operator scenarios. They enable secure proactive handover services not only for the users of the WMN, but also for parts of the network infrastructure. We proposed three protocols that can be used by devices at different stages of their network usage, e.g., during the initial EAP authentication, or any time after. The protocols Neighborhood Pre-Authentication (NPA) and EAP-TTLS Neighborhood Pre-Authentication (ENPA) allow to bootstrap multiple Target Mesh Router (TMR) at once, while the 3-Party Handshake for Handover (3PHSH) protocol prepares exactly one TMR per protocol run. All protocols are compatible to the FAU-concept, as well as the multi-operator generalization (cf. Chapter 6).

Concluding, this thesis proposed a comprehensive security architecture for WMNs interfacing with standardized mechanisms such as IEEE 802.11i, EAP, RADIUS, and IPsec. The core feature is an extensible key hierarchy that is used to set up security associations between network nodes after they have been successfully authenticated. The key hierarchy can also be used in the context of multiple operators, whether they cooperate in forming a mixed network, or maintain separate networks. We also proposed three proactive handover protocols for WMNs that are secured with the help of the security associations bootstrapped using the FSASD key hierarchy.
In this section we shortly elaborate on open issues that require additional research outside of the scope of this thesis. This includes communication between AAA servers in multi-operator scenarios, handover decisions, and the testbed evolution.

Chapter 6 detailed the generalization of the FSASD security architecture to support multiple operators including MC roaming. Most scenarios involving multiple operators require communication between their AAA servers, e.g., during the authentication of roaming devices in a Foreign Network (FN). It should further be investigated how the communication between AAA server, i.e., the key transport from one domain to another, can be secured. Proposed mechanisms such as [HNO10], [RRSW97] or [SDNN08] are not specific enough to allow a standardized secure key transport, yet. In terms of the FSASD key hierarchy, it should be considered whether further extensions may aid in this endeavor. It may for instance be beneficial for operators to cross-authenticate their AAA servers or their MGs representing the network perimeter based on the FSASD mechanisms.

As Chapter 7 has shown, a secure efficient and practical handover can be achieved using the FSASD key hierarchy. The evaluation of our three handover protocols has also shown (cf. Figure 7.18) that not only the protocols themselves influence the overall handover time, but also driver related issues such as scanning the spectrum. In our research we focused on secure key transport, however, the decision when to initiate a handover remains an open issue. It requires scenario-specific research into wireless signal reception and most importantly movement prediction of the device requiring handover. There is also much room for improvement on the wireless device drivers of the Linux operating system which we used during our evaluations.

Last, the ITsec testbed (cf. Chapter 4) as a platform to develop and evaluated security protocols for WMNs should be evolved constantly. Besides keeping up with the state of the art, more effort should be spent on further automating the testbed. This includes our monitoring and management solution pwrmesh, but also the means to automate tests and evaluations of new protocols.
Acronyms

3PHSD  3-Party Handshake for Sequential Deployment  
3PHSH  3-Party Handshake for Handover  
AAA  Authentication, Authorization and Accounting  
AC  Access Controller  
AH  Authentication Header  
AKCK  Abbreviated Handshake Key Confirmation Key  
AKEK  Abbreviated Handshake Key Encryption Key  
ANQP  Access Network Query Protocol  
AODV  Ad-hoc On Demand Distance Vector  
AP  Access Point  
AVISPA  Automated Validation of Internet Security Protocols and Applications  
AVP  Attribute Value Pair  
BSS  Basic Service Set  
BSSID  Basic Service Set Identifier  
CA  Certification Authority  
CAPWAP  Configuration And Provisioning for Wireless Access Points  
CMR  Current Mesh Router  
DMA  Direct Memory Access  
DoS  Denial of Service  
DSRK  Domain Specific Root Key  
DSUSRK  Domain Specific Usage Specific Root Key  
EAP  Extensible Authentication Protocol  
EAPOL  EAP Transport Over LAN  
EAP-TLS  EAP Transport Layer Security  
EAP-TTLS  EAP-Tunneled Transport Layer Security  
ECC  Elliptic Curve Cryptography
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMSK</td>
<td>Extended Master Session Key</td>
</tr>
<tr>
<td>ENPA</td>
<td>EAP-TTLS Neighborhood Pre-Authentication</td>
</tr>
<tr>
<td>ERP</td>
<td>EAP re-Authentication Protocol</td>
</tr>
<tr>
<td>ESP</td>
<td>Encapsulating Security Payload</td>
</tr>
<tr>
<td>ESS</td>
<td>Extended Service Set</td>
</tr>
<tr>
<td>FAU</td>
<td>FSASD Authenticator</td>
</tr>
<tr>
<td>FDRK</td>
<td>FAU Domain Root Key</td>
</tr>
<tr>
<td>FN</td>
<td>Foreign Network</td>
</tr>
<tr>
<td>FO</td>
<td>Foreign Operator</td>
</tr>
<tr>
<td>FSASD</td>
<td>Framework for establishing Security Associations for Sequentially Deployed WMN</td>
</tr>
<tr>
<td>GAS</td>
<td>Generic Advertisement Service</td>
</tr>
<tr>
<td>GMSK</td>
<td>Group Master Session Key</td>
</tr>
<tr>
<td>GTK</td>
<td>Group Transient Key</td>
</tr>
<tr>
<td>HMAC</td>
<td>Keyed-Hash Message Authentication Code</td>
</tr>
<tr>
<td>HN</td>
<td>Home Network</td>
</tr>
<tr>
<td>HO</td>
<td>Home Operator</td>
</tr>
<tr>
<td>IBC</td>
<td>Identity-based Cryptography</td>
</tr>
<tr>
<td>IBSS</td>
<td>Independent Basic Service Set</td>
</tr>
<tr>
<td>ID</td>
<td>Identification</td>
</tr>
<tr>
<td>IEEE</td>
<td>IEEE</td>
</tr>
<tr>
<td>IE</td>
<td>Information Element</td>
</tr>
<tr>
<td>IKE</td>
<td>Internet Key Exchange Protocol</td>
</tr>
<tr>
<td>IKEv2</td>
<td>Internet Key Exchange Protocol v2</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPsec</td>
<td>Internet Protocol Security</td>
</tr>
<tr>
<td>ISAKMP</td>
<td>Internet Security Association and Key Management Protocol</td>
</tr>
<tr>
<td>KDF</td>
<td>Key Derivation Function</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LOS</td>
<td>Line of Sight</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>rTK</td>
<td>Root Traffic Key</td>
</tr>
<tr>
<td>RTT</td>
<td>Round Trip Time</td>
</tr>
<tr>
<td>SAE</td>
<td>Simultaneous Authentication of Equals</td>
</tr>
<tr>
<td>SA</td>
<td>Security Association</td>
</tr>
<tr>
<td>SCTP</td>
<td>stream control transport Protocol</td>
</tr>
<tr>
<td>SNMP</td>
<td>Simple Network Management Protocol</td>
</tr>
<tr>
<td>SPI</td>
<td>Security Policy Identifier</td>
</tr>
<tr>
<td>SSID</td>
<td>Service Set ID</td>
</tr>
<tr>
<td>STA</td>
<td>Station</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport Control Protocol</td>
</tr>
<tr>
<td>TEK</td>
<td>Traffic Encryption Key</td>
</tr>
<tr>
<td>TFTP</td>
<td>Trivial File Transfer Protocol</td>
</tr>
<tr>
<td>TIK</td>
<td>Traffic Integrity Key</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>TMR</td>
<td>Target Mesh Router</td>
</tr>
<tr>
<td>TTP</td>
<td>Trusted Third Party</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UID</td>
<td>Unique Identifier</td>
</tr>
<tr>
<td>USRK</td>
<td>Usage Specific Root Key</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over IP</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless LAN</td>
</tr>
<tr>
<td>WMN</td>
<td>Wireless Mesh Network</td>
</tr>
<tr>
<td>WTP</td>
<td>Wireless Termination Point</td>
</tr>
</tbody>
</table>
Bibliography


[IEE07] , *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-
WPANs): Amendment to add alternate PHY, IEEE Standard 802.15.4, 2007. (Cited on page 54.)


Andreas Meissner, Thomas Luckenbach, Thomas Risse, Thomas Kirste, and Holger Kirchner, *Design Challenges for an Integrated Disaster Management Communication and Information System*, International Conference on Computer Communications (INFOCOM), IEEE, 2002. (Cited on pages 48 and 53.)


Mark S. Miller, Ka-Ping Yee, and Jonathan Shapiro, *Capability Myths Demolished*, Tech. report, Systems Research Laboratory, Johns Hopkins University, 2003. (Cited on page 40.)


[SHJB07] H. Skinnemoen, S. K. Hansen, A. Jahn, and M. Berioli, Satellite Based Infrastructure for Emergency Communications, International Commu-


