Lehrstuhl für Kommunikationsnetze
Prof. Dr.-Ing. Christian Wietfeld

SECURE AND EFFICIENT ROUTING IN HIGHLY DYNAMIC WLAN MESH NETWORKS

technische universität
dortmund

DISSERTATION

For the Degree of *Doktor-Ingenieur*
Faculty of Electrical Engineering and Information Technology
TU Dortmund University, Germany

Mohamad Sbeiti
Dortmund, September 2015

Author's Contact Information:
mohamad.sbeiti@tu-dortmund.de
www.paser.info

Thesis Advisor:	**Prof. Dr.-Ing. Christian Wietfeld** TU Dortmund University
Secondary Referee:	**Prof. Dr. Thorsten Holz** Ruhr University Bochum
Thesis Submitted:	April 23, 2015
Thesis Defense:	August 21, 2015

To all people deprived of education.

To Kassem, Sanaa, Malak, Hiyam, Moussa and May
for their endless love and support.

Acknowledgment

Praise be to Allah, lord of the worlds. The beneficent, the merciful.

My utmost gratitude goes to my supervisor, Prof. Dr.-Ing. Christian Wietfeld, for giving me the opportunity to join his group, for letting me participate in several research projects, and for securing financial support to finish this work. He is a professional advisor, from whom I learned not only how to do research, but also to embrace challenges and to always strive for the best. I thank him for his continuous and consistent support, guidance, and encouragement. My discussions with him, even though sometimes difficult, repeatedly inspired me to tackle problems from different and greater perspectives.

I gratefully acknowledge Prof. Dr. Thorsten Holz, Prof. Dr.-Ing. Peter Krummrich, and Prof. Dr.-Ing. Christian Rehtanz for serving as my supervisory committee and for helping me to improve the quality of this thesis by their valuable comments and feedback.

Being part of the communication networks institute (CNI) at the TU Dortmund gave me the opportunity to collaborate with many talented researchers. I would like to express my deep gratitude to all those on the CNI research team, especially Dipl.-Inf. Maike Kuhnert, Dr.-Ing. Andreas Wolff, Dipl.-Ing. Sebastian Subik, and Dr.-Ing. Thang Tran for always finding time whenever I needed an intelligent discussion. Working with them for many years enabled me to conduct quality research. I also thank Dipl.-Inf. Niklas Goddemeier, Dipl.-Inf. Daniel Behnke, Dipl.-Ing. Christoph Ide, and M.Sc. Sebastian Rohde for our successful joint work on several projects and papers. I have greatly benefited from the many interactions and discussions with them.

I am also grateful to the department of electrical engineering and information technology at the Ruhr University Bochum, especially to Prof. Dr.-Ing. Christof Paar, Prof. Dr. Jörg Schwenk, and Dr. Roberto Avanzi, for teaching me IT security, which has been the basis of this thesis.

I also would like to acknowledge Dr.-Ing. Shadi Traboulsi, Dr. Ahmed E.A.A. Abdulla, Dipl.-Ing. Carsten Vogel, Dr. Houssein Assaad, Dr.-Ing. Hassan Sbeyti, Dr. Hassan Hijazi, and Dipl.-Ing. Jakob Pojda for reviewing some papers of mine and for the fruitful discussions we had about different research topics. They always had the kindness to give helpful comments.

I strongly acknowledge the code guru, Dipl.-Ing. Eugen Paul, and Dipl.-Ing. Carsten Vogel for supporting me in implementing my research project, the secure

routing protocol PASER, in simulation and in Linux. I would not have been able to study secure routing in wireless mesh networks as thoroughly without their support. I also thank B.Sc. Jan Schröder, B.Sc. Majuran Rajakanthan, Dipl.-Ing. Mohamad Nehme, and M.Sc. Jonas Hinker for helping me to tackle different security aspects of wireless mesh networks.

On a more personal note, I am deeply indebted to my friends who have always stood by my side. I thank my relatives for their cherish and unwavering support. Gratefully, I thank my parents for their sacrifices, continuous support, care, guidance, and encouragement. I dedicate each and every one of my successes to them. I also thank my sister, May, and my brother, Moussa, for their endless love. Finally, I would like to thank my wife, Hiyam, who cherished me through difficult times and comforted me in stressful situations. Without her genuine love and companionship, I would have taken much longer to finish this work.


overhead
Linux
Message
MAC
ARAN
BATMAN
Communications
operations
time
proactive
Packet
Wireless
mechanisms
routing
Link
HWMPS
packets
Time
Dynamic
Simulation
function
performance
scenario
Available
mobility
WLAN
range
PASER
messages
mobile
scenarios
process
first
Evaluation
discovery
communication
WMNs
Node
practice
frames
Protocol
International
Protocols
key
destination
dynamic
results
set
traffic
access
Performance
information
module
Networks
link
packet
reactive
model
INETMANET
routes
protocols
attacker
topology
number
attack
delay
authentication
transmission
attacks
symmetric
mechanism
volume
Security
security
Implementation
IEEE
HWMP
kernel
802.11s
gateway
use
Route
wireless
links
neighbours
one-hop
chain
simulation
PHY
channel
secure
Online
network
Highly
static
WMN
position
OMNeT++
different
Mobile
Mesh
size
corresponding
approach
throughput
OLSR
data
Mbit/s
Secure
Analysis
rate
standard
scheme
implementation
Key
Network
networks
highly
Conference
framework
route-o-matic
due
cryptographic
hash
UAVs

Abstract

Recent advances in embedded systems, energy storage, and communication interfaces, accompanied by the falling prices of WLAN routers and a considerable increase in the throughput of a WLAN (IEEE 802.11), have facilitated the proliferation of WLAN Mesh Network (WMN) applications. In addition to their current deployments in less dynamic community networks, WMNs have become a key solution in various highly dynamic scenarios. For instance, WMNs are intended to interconnect self-organized, cooperative, and small Unmanned Aerial Vehicles (UAVs) in a wide range of applications, such as emergency response, environmental monitoring, and ad-hoc network provisioning. Nevertheless, WMNs still face major security challenges as they are prone to routing attacks. Consequently, the network can be sabotaged and, in the case of UAV-WMN-supported missions, the attacker might manipulate payload data or even hijack UAVs.
Contemporary security standards, such as the IEEE 802.11i and the security mechanisms of the IEEE 802.11s mesh standard, are vulnerable to routing attacks, as experimentally shown in this research. Therefore, a secure routing protocol is indispensable for making feasible the deployment of WMNs in critical scenarios, such as UAV-WMN-assisted applications. As far as the author of this thesis knows, none of the existing research approaches for secure routing in WMNs have gained acceptance in practice due to their high overhead or strong assumptions.
In this research, a new approach, which is called Position-Aware, Secure, and Efficient mesh Routing (PASER), is proposed. This new proposal defeats more attacks than the IEEE 802.11s/i security mechanisms and the well-known, secure routing protocol Authenticated Routing for Ad-hoc Networks (ARAN), without making restrictive assumptions. It is shown that PASER achieves —in realistic UAV-WMN scenarios— similar performance results as the well-established, non-secure routing protocols Hybrid Wireless Mesh Protocol (HWMP) combined with the IEEE 802.11s security mechanisms. Two representative scenarios are considered: (1) on-demand ubiquitous network access and (2) efficient exploration of sizable areas in disaster relief. The performance evaluation results are produced using an experimentally validated simulation model of WMNs, realistic mobility patterns of UAVs, and an experimentally derived channel model for the air-to-air WMN link between UAVs. The findings of this evaluation are justified by the route discovery delay and the message overhead of the considered solutions.

This research's main aim is to support the broad deployment of a secure routing protocol in highly dynamic WMNs, and the PASER features, presented here, are a major step towards achieving this goal.

Kurzfassung

Der technologische Fortschritt in verschiedenen Bereichen wie eingebetteten Systemen, Energiespeicherung sowie Kommunikationsschnittstellen und -protokollen, hat, unterstützt von sinkenden Preisen für WLAN-Hardware und dem deutlichen Anstieg der Übertragungsraten von WLAN-Technologien (IEEE 802.11), den Grundstein gelegt für die zunehmende Anzahl an Anwendungen von WLAN-Mesh-Netzen (WMNs). Diese haben sich neben der gegenwärtigen Verbreitung quasistatischer, Community-gestützter Funknetze vor allem zu einer Schlüsseltechnologie für hochdynamische Anwendungen entwickelt. Eine solche hochdynamische Anwendung ist beispielsweise die WMN-gestützte Kommunikation selbstorganisierender, miteinander kooperierender unbemannter Kleinflugkörper (UAVs) mit einer grossen Spanne an Einsatzbereichen, wie Bergungs- und Rettungseinsätze, Umweltüberwachung, Präzisionslandwirtschaft oder der Ad-hoc-Bereitstellung mobiler WLAN- und Mobilfunknetze. Mit zunehmendem Einsatz in solch kritischen Bereichen wachsen auch die Anforderungen an die Sicherheit. Gegenwärtige WMN-Technologien stehen hierbei noch vor grossen Sicherheitsherausforderungen, da diese keinen Schutz gegen Routing-Angriffe gewähren. Mithin besteht das Risiko, dass Angreifer Netze sabotieren und im Falle von UAV-Einsätzen Nutzdaten der UAVs manipulieren oder sogar die gesamte Kontrolle über einzelne UAVs übernehmen können.

Wie in dieser Dissertation gezeigt wird, sind derzeitige Sicherheitsstandards wie IEEE 802.11i oder auch die Sicherheitsmechanismen des Mesh-Standards IEEE 802.11s anfällig für Routingangriffe. Um einen sicheren Einsatz von WMNs im sensiblen und kritischen Umfeld wie den erwähnten UAV-WMN-basierten Anwendungsszenarien zu gewährleisten, muss auch das eingesetzte Routingprotokoll entsprechend sicher sein. Soweit dem Autor bekannt, konnte sich unter diesem Gesichtspunkt keine der existierenden Lösungen in der Praxis durchsetzen, da entweder der Overhead der verschiedenen Ansätze zu hoch oder aber die getroffenen Annahmen in Sinne der Sicherheit zu stark und damit in der Praxis lediglich begrenzt umsetzbar sind.

In dieser Arbeit wird ein alternativer, umfassender Ansatz namens Position-Aware, Secure and Efficient mesh Routing (PASER) vorgestellt. Dieses neue Routingverfahren bietet eine höhere Sicherheit gegen Routingattacken als die bestehenden IEEE 802.11s/i-Sicherheitsmechanismen und sogar als das sichere Routingprotokoll Authenticated Routing for Ad-hoc Networks (ARAN), wobei allerdings im Hinblick auf die praktische Einsetzbarkeit auf starke Annahmen verzichtet wird. Es wird gezeigt, dass PASER in realistischen UAV-WMN Szenarien eine vergleichbar hohe Performanz erreicht, wie sie mit dem derzeit weit verbreiteten, aber unsicheren Routingprotokoll Hybrid Wireless Mesh Protocol (HWMP) in Kombination mit den IEEE 802.11s Sicherheitsmechanismen

möglich ist. Zwei repräsentative Szenarien werden hierfür betrachtet: (1) ubiquitärer ad-hoc Netzzugriff und (2) die effiziente Erkundung grösserer Gebiete im Katastrophenschutz. Zur Leistungsbewertung werden ein durch Experimente validiertes OMNeT++-basiertes WMN-Modell mit wirklichkeitsgetreuen UAV-Mobilitätsmustern und eben falls ein aus Experimenten abgeleitetes Kanalmodell für die direkte Funkverbindung zwischen UAVs herangezogen. Bei der Analyse der Ergebnisse werden auch die Verzögerungen der Routenfindung und der Nachrichten-Overhead der Protokolle berücksichtigt.

Diese Arbeit und das in diesem Rahmen erforschte PASER-Verfahren sollen einen wesentlichen Beitrag leisten auf dem Weg zur Entwicklung sicherer, praxistauglicher Routingprotokolle mit dem Ziel einer möglichst hohen Verbreitung eines solchen Protokolls in hochdynamischen WMNs.

Contents

1

Introduction

Contents of this Chapter

In this chapter, the background for conventional, less dynamic WLAN mesh networks is given, followed by the motivation for the need for highly dynamic WLAN mesh networks. The necessity of secure routing in highly dynamic networks is then discussed. Afterwards, the main contributions of this research are elaborated, and the organization of the thesis is presented.

1.1 Background for Conventional WLAN Mesh Networks

Over the last decade, multi-hop wireless communication networks have received increased attention due the growing demand for low-cost and high-performance ubiquitous networking. In the late 1970s, the first generation of Mobile Ad-hoc NETworks (MANETs) was proposed [Kah78], and by the early 1990s, MANETs had become a key solution for military applications and emergency operations [Mac98]. MANETs are infrastructureless and non-hierarchical multi-hop wireless networks consisting solely of mobile nodes. That is, in a MANET, the mobile nodes set up and maintain a network on demand without any support from an existing infrastructure. In this way, all the nodes have identical responsibilities, including the routing and forwarding of data. MANETs have the advantages of being self-forming, auto-configuring, and self-healing (automatic configuration

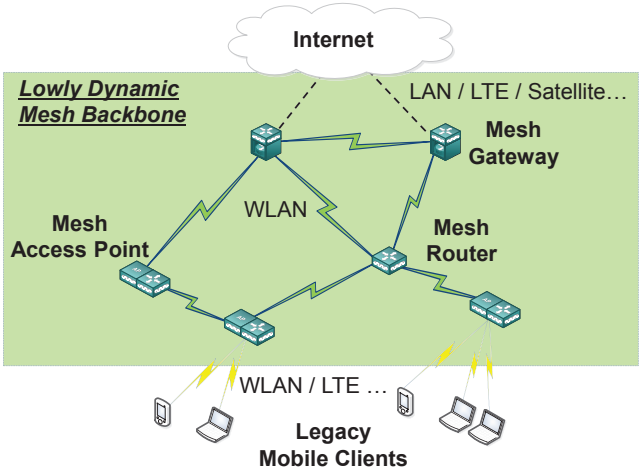

Figure 1.1: Architecture of a conventional WLAN mesh network.

and routing). Despite these advantages, MANETs have not gained wide acceptance in civilian applications. One primary reason could be the low incentive of users to forward the data of others using their power-constrained mobile devices. According to [Wan09], two other reasons are concerns about security and about the limited throughput of a MANET. In the early 2000s, Wireless Local Area Network (WLAN) mesh networks (WMNs) emerged [Aky05] to address the limitations of MANETs and to offer a more promising multi-hop wireless communication solution for civilian applications. Figure 1.1 illustrates the architecture of a conventional WLAN mesh network. As the figure shows, WMNs have, in contrast to MANETs, a hierarchical structure. The nodes have different functionalities rather than being equal peers. WMNs are mainly composed of legacy mobile clients (WLAN or Long Term Evolution (LTE) clients or others) and the mesh backbone. The latter is dedicated to network configuration and routing. It offers, on demand, network coverage to the clients, and it deals with the transparent delivery of their data. The mesh backbone comprises mesh routers, mesh access points, and mesh gateways. Mesh routers are wireless relays which run a routing protocol to dynamically set up and maintain routes in the network. Mesh access points are mesh routers that also provide network access to clients. Mesh gateways are mesh routers that connect the network to the Internet. That is, in contrast to MANETs, WMNs rely on existing infrastructures to offer Internet connectivity to clients. Besides, in a conventional WMN, the mesh backbone has no mobility [IEE11, Wan09, Aky05]. The mesh backbone

Table 1.1: Main differences between conventional WMNs and MANETs.

WMNs	**MANETs**
Hierarchical network	Flat network
Rely on infrastructure	No infrastructure
Less dynamic topology (No mobility in mesh backbone)	Highly dynamic topology (Mobile nodes)
Grid-connected mesh backbone nodes	Power-constrained nodes
Multiple radios and multiple channels	Single radio and single channel
Main traffic: user↔gateway	User↔user traffic

nodes are placed in fixed positions (e.g., the roof of a building). These are also not power-constrained, as are the mobile devices in MANETs. They are generally grid-connected. In addition, the nodes in the mesh backbone might comprise multiple radio interfaces and can operate using multiple orthogonal channels. For instance, mesh access points typically have one radio interface to perform routing between the backbone nodes, and another radio interface to offer network access for clients. Another characteristic of WMNs is that most of the traffic is destined for the gateway (Internet), while the traffic in MANETs is mainly between temporary node pairs. Table 1.1 illustrates the main differences between conventional WMNs and MANETs.
Due to the inherent advantages of WMNs, such as high throughput per link (up to 800 Mbit/s in the case of 2x2 Multiple Input Multiple Output (MIMO) and channel bundling (40 MHz) [IEE13a]), improved network capacity in the case of multiple orthogonal channels over multiple interfaces, coverage of sizable areas, Internet connectivity, and cost-efficient and flexible deployment [Aky05], WMNs are witnessing increased deployments in a wide range of applications. Example applications include community networks, such as Freifunk Berlin [FRE] and MIT Roofnet [Bic05a], network provisioning in large areas, such as Google WiFi [GOO] and VillageNet [Dut07], smart metering [Lic10], and emergency operations [Wol12].

Having the aforementioned applications in mind, the majority of existing research on WMNs have focused on a less dynamic mesh backbone, which is a typical characteristic in those applications. Less dynamic means that the topology of the backbone only changes due to the addition or removal of a node or due to changes in the network environment. In contrast, in this thesis, highly dynamic mesh backbones are addressed. Here, the topology also changes due to the mobility of the nodes. The motivation for tackling this evolving kind of network is covered in the next section.

1.2 The Need for Highly Dynamic WLAN Mesh Networks

In recent years, small, low-altitude Unmanned Aerial Vehicles (UAVs) have become available on the market at affordable prices [Aus10]. With a weight of less than 1 kg, UAVs can be equipped with a variety of sensors, wireless transceivers, and positioning capabilities. They can be used for the cooperative exploration of dangerous scenarios to avoid risking the health and lives of personnel. With a diameter below 1 m, they can be deployed in terrain that is difficult to reach with conventional vehicles. Supporting an autonomous behavior and speeds up to 20 m/s, they have the ability to complete missions in shorter periods of time than classical means. In addition, they can be used as communication hotspots or relays to assist (or replace) existing networks. This has rendered the deployment of cooperative UAVs in a wide range of applications very attractive. UAV-assisted applications include, but are not limited to, delivering Internet to the third-world [FAC], coverage extension or densification [Abd12, Roh13], disaster relief [Dan09], polar weather monitoring [Cur04], and precision farming [Tec08]. Nevertheless, for such applications to become a reality, a reliable, auto-configuring, and self-healing wireless backbone network is needed to interconnect the UAVs and to provide a connection to their ground control station, the cellular core network, or/and the Internet. WMNs are a good candidate, as they have the aforementioned characteristics [Aky05], and they offer a physical air-to-air link for a direct communication between the UAVs. Hereafter, such a network is referred to as an UAV-WMN. As a significant difference from the ground-based WMNs that are less dynamic, i.e., the topology of the backbone only changes due to the addition or removal of a node or due to changes in the network environment, the topology of an UAV-WMN also changes due to mobility. The UAV-WMN nodes feature relative speeds between 15 and 26 m/s [Asa13] leading to high topology change rates, up to 0.1 Hz [Sbe15]. That is, the topology of an UAV-WMN is highly dynamic. However, the non-inherent mobility of nodes is not random and non-deterministic. It is rather a controlled mobility that enables establishing highly efficient network topologies. Moreover, by applying communication-aware mobility to the UAVs, it becomes possible to react to time varying channel conditions and topology changes in real time [Wie14]. Hence, self-organization capabilities like node and relay placement, connectivity restoration, RF-signal outage protection and compensation can be exploited to establish reliable UAV-WMN in highly dynamic and complex environments.

In this thesis, the focus lies on UAV-WMN-assisted disaster relief as a reference scenario. In this context, the recent United Nations' global assessment report on disaster risk reduction [UN] reveals an increase in the number of disasters in the last years that resulted in severe humanitarian disasters and economic damage. The report indicates that one of the top concerns in disaster areas is the disruption of telecommunications. Sugino [Sug12] reports, in a summary of

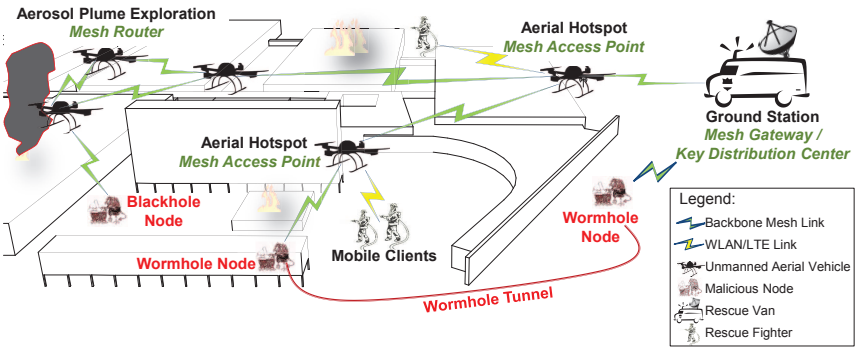

Figure 1.2: Example of a deployment scenario of UAV-WMN and two routing attacks in disaster relief [Sbe15].

the damages of the great east Japan earthquake and tsunami in March 2011, that 1.9 million fixed telephone lines and 29,000 cellular base stations were damaged. He also reveals that emergency restoration of communication networks took one month, while a full restoration took 11 months. These facts emphasize the increasing importance of portable communication networks in disaster areas. Moreover, these figures point out that a communication network that can be deployed in a substantially short period (e.g., one hour) is indispensable to efficiently cope with large scale crises. UAVs connected via a WMN (UAV-WMN) and acting as WLAN or LTE aerial hotspots meet these requirements [FAC, Roh13]. In addition, an UAV-WMN is indispensable in such scenarios to efficiently explore areas where uncontrolled emissions of liquid or gaseous contaminants exist [God12a, Dan09]. Figure 1.2 illustrates how an airborne mesh network consisting of UAVs connected via a WMN can be used to assist disaster relief operations. As the figure shows, UAVs build a portable wireless mesh backbone. This backbone offers, on demand, network coverage to legacy mobile WLAN/LTE clients (rescue fighters' devices), and it deals with the transparent delivery of their data. Besides, the UAV-WMN backbone is used to transmit the sensor information of a group of UAVs performing exploration tasks.

From a routing point of view, highly dynamic WMNs frequently lead to broken links, to which a routing protocol should quickly react. From a security point of view, (re-)authentication occurs very often, thus, this step should be efficiently managed. Besides, the consistent reachability of a central security server is very challenging in highly dynamic WMNs, thus, a security scheme should avoid relying on that.

1.3 Problem Statement: Routing Security Issues

While the WMN capability for auto-configuration and self-healing significantly reduces the complexity of network deployment and maintenance, it makes the WMN backbone prone to routing attacks, which include the blackhole and wormhole attacks [Kan07]. As a result, the attacker can, with little cost and effort, redirect the traffic and drop the data packets even if the wireless backbone links are encrypted. In UAV-WMN-assisted disaster relief situations, this can sabotage the communication between rescue fighters. In addition, the command and control data exchanged between the UAVs and their ground station will get disrupted. This issue makes the use of WMNs (or any wireless multi-hop solution relying an a routing protocol to dynamically set up routes) problematic for the command and control of UAVs in practice, as flight regulations impose that it should be always possible to remotely pilot the UAVs [FAA]. Because the UAVs are highly dynamic, relying on the exchange of information for autonomous cooperative positioning [God12a], the attacker might also alter their flight paths by selectively dropping packets. In case the attacker is able to compromise network credentials and as long as there is no efficient way to refresh those credentials, the attacker might manipulate payload data or even inject corrupted control information that could lead to the highjacking of an UAV. For instance, the attacker might impersonate an UAV and propagate corrupted position information, exploiting the UAVs' collision avoidance mechanisms to indirectly steer UAVs to areas controlled by the attacker. Since the disruption of communications and the violation of the flight security of UAVs can lead to fatal consequences (e.g., near airports), it is vital to deploy a secure UAV-WMN backbone. Two approaches to secure the communication between mesh backbone nodes exist:

1. Combining well-established, non-secure routing protocols with standardized security mechanisms, such as those of the Institute of Electrical and Electronics Engineers (IEEE) association: The IEEE 802.11i [IEE04] security standard or the recent IEEE 802.11s [IEE11] mesh standard. However, as shown in this research, these standards are vulnerable to the blackhole and wormhole attacks.

2. Use of a secure mesh routing protocol. Many secure routing research approaches have been proposed in the last decade [Sen13, Sgo13, Abu08, Hu04], but none of them has been deployed in practice. The high overhead of the security mechanisms of these protocols or the strong assumptions made during their design (e.g., the existence of an efficient symmetric key management scheme) have rendered their deployment in real life applications infeasible.

Although there are already practical projects involving wireless mesh networks [Ulu12], their security has been, in practice, given only marginal attention. It has been considered as a challenge for the next-generation of wireless mesh networks [Sen13].

1.4 Thesis Contributions

A considerable amount of ongoing research dealing with crisis management optimization has been focusing on the development of a deployable secure mesh routing protocol [AIR, ANC, Büs14]. This thesis makes the following noteworthy contributions:

- The Position-Aware Secure and Efficient mesh Routing approach (PASER) is presented. PASER uses a hybrid cryptosystem to address the overhead problem of approaches based on asymmetric cryptosystems. Besides, to tackle the dependency of symmetric cryptosystem-based proposals on a key distribution method, which in turn requires either secure routes or secure out-of-band channels, PASER incorporates an in-band key management scheme. This allows for rapid response to security breaches, and resolves a major issue in current deployments [Leb12].

- An accurate (close to reality) implementation of PASER in the discrete event-based network simulator OMNeT++ [Var08] is provided. This implementation has been integrated in the official INETMANET framework [PAS] as a first reference implementation for secure mesh routing protocols. The INETMANET framework comprises comprehensive simulations models of MANETs, WMNs, and standard network protocols.

- The feasibility of PASER is proven by contributing a Linux-based implementation of PASER in a WMN testbed. This implementation is composed of a novel Linux kernel framework designed for reactive routing protocols in general and a PASER-specific Linux user space implementation.

  A website about PASER (www.paser.info) has been released. The PASER codes in simulation and in Linux, and a thorough documentation of these codes, are posted as open sources on this website. Reusing these codes should ease the development and evaluation of new routing protocols. Besides, these codes provide a reference for the comparison of secure routing protocols, which is currently missing in both in simulation and in practice.

- A security analysis as well as an extensive performance evaluation of PASER and the following three representative alternate solutions are provided:

1. ARAN: The well-known, reactive, and secure routing protocol Authenticated Routing for Ad-hoc Networks (ARAN) [San05].

2. HWMPS: A combination of the security mechanisms of the IEEE 802.11s mesh standard and the Hybrid Wireless Mesh Protocol (HWMP), which is specified in the mentioned standard.

3. BATMANS: A combination of the IEEE 802.11i security mechanisms and the Better Approach To Mobile Ad-hoc Networking (BATMAN) proactive routing protocol [BATa], which is widely deployed in community networks [BATc].

– It is shown, in a WMN testbed, that the IEEE 802.11 security frameworks are vulnerable to the blackhole and wormhole attacks, and that PASER mitigates these attacks. Besides, based on a theoretical analysis, the results show that PASER mitigates —in UAV-WMN— more attacks than the three alternatives.

– The route discovery delays of PASER and the three alternate solutions are analyzed in theory and in simulation. Lower bound equations for these delays are derived as they constitute along with the routing overhead, for which asymptotic expressions are provided, the main impact on the overall network performance. The results show that PASER (also HWMP) has a more efficient and robust route discovery process than ARAN and BATMANS, and it is scalable with respect to network size and traffic load.

– Using the network simulator OMNeT++, realistic UAV-mobility patterns, and an experimentally derived channel model for the air-to-air UAV-WMN links, the performance of the protocols is investigated in different UAV-WMN scenarios under multiple traffic types and various scenario sizes. The results show that PASER achieves —in the investigated scenarios— performance comparable to that of HWMPS.

This combination of values in PASER (security and performance) is deemed to be necessary by the Internet Engineering Task Force (IETF) keying and authentication for routing protocols group [IETa] to drive a broad deployment of a secure routing protocol.

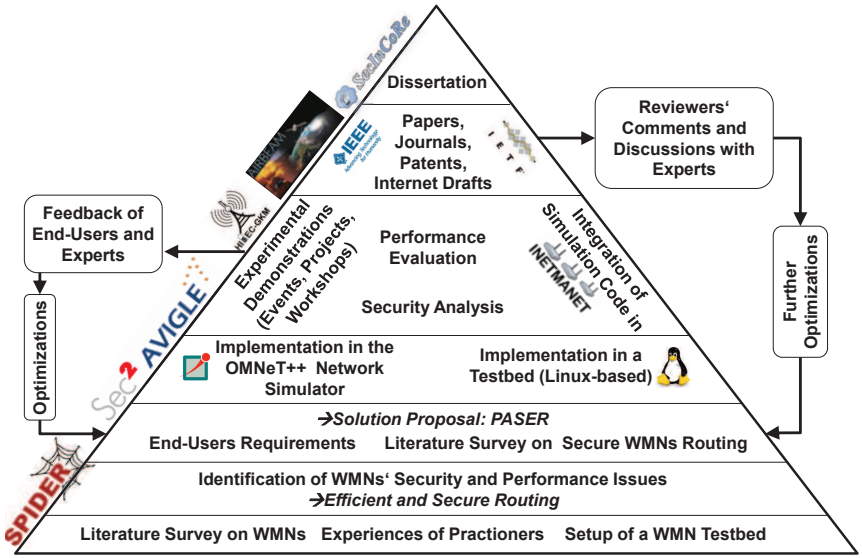

Figure 1.3: Research methodology of this thesis.

1.5 Thesis Methodology

The research methodology applied in this thesis is depicted in Figure 1.3. First, the following steps have been taken to identify the performance and security issues of (UAV-)WMNs, including the secure routing problem:

- The experiences and recommendations of WMN stakeholders, such as in [AIR, Roh10, Šub10], regarding the white spots and performance bottlenecks of (UAV-)WMNs.
- A literature survey (e.g., books, journals, papers) on (UAV-)WMNs and their current issues.
- Lessons learned from a (UAV-)WMN testbed developed at the Communication Networks Institute (CNI) of the TU Dortmund University.

Having identified several (UAV-)WMNs challenges, such as efficient rate selection algorithms, interference-resilient channel medium access, and secure routing, the focus in this thesis lies on the last, as secure routing is a fundamental need to make feasible the deployment of WMNs. To fill

the existing secure routing gap in UAV-WMN, PASER is proposed in this research, after having accomplished the following steps:

- Determining the advantages and drawbacks of existing secure routing solutions based on an extensive literature survey (e.g., books, journals, papers, IETF documents).
- Eliciting end-user requirements within several interdisciplinary research projects, such as SecInCoRe [SEC], AIRBEAM [AIR], AVIGLE [Roh10], and SPIDER [Šub10]. The contacted end-users include, but are not limited to, the fire brigades of Gelsenkirchen, the German Red Cross, and the police of the North Rhine-Westphalia.

For an extensive performance evaluation of the proposed solution, PASER is implemented in an experimentally-validated OMNeT++-based UAV-WMN simulation model. In addition, as it is striven for a deployable solution in this research, the feasibility of PASER is proved using a Linux-based real implementation. The corresponding scientific results have been published in peer-reviewed conference proceedings and a journal submission. The comments of reviewers as well as the results of fruitful discussions with the audience have led to further optimizations of the proposed approach. More than 12 peer-reviewed scientific publications and one patent application show the positive response to the applied methodology. Solutions based on PASER were presented as live demonstrations in several events and workshops, among others, the Vodafone innovation days in Düsseldorf, Germany in November 2014 [Sbe14a] and the SecInCoRe end-user workshop in Manchester, England in December 2014 [Kuh15]. The feedback of the demonstrations always emphasized the added value of the proposed approach.

1.6 Thesis Outline

An overview of the organization of this thesis and the core publications of this research are depicted in Figure 1.4. The rest of this thesis is structured as follows:

- Chapter 2 gives an introduction to the different routing classes in wireless multi-hop networks and covers the most-established (non-secure) routing protocols of each class, that are relevant for highly dynamic WMNs. These protocols are used as references to evaluate the performance overhead of the secure routing approach proposed in this thesis.
- Chapter 3 explores the shortcomings of the IEEE 802.11 s/i security mechanisms to secure the backbone of highly dynamic WMNs. The chapter also

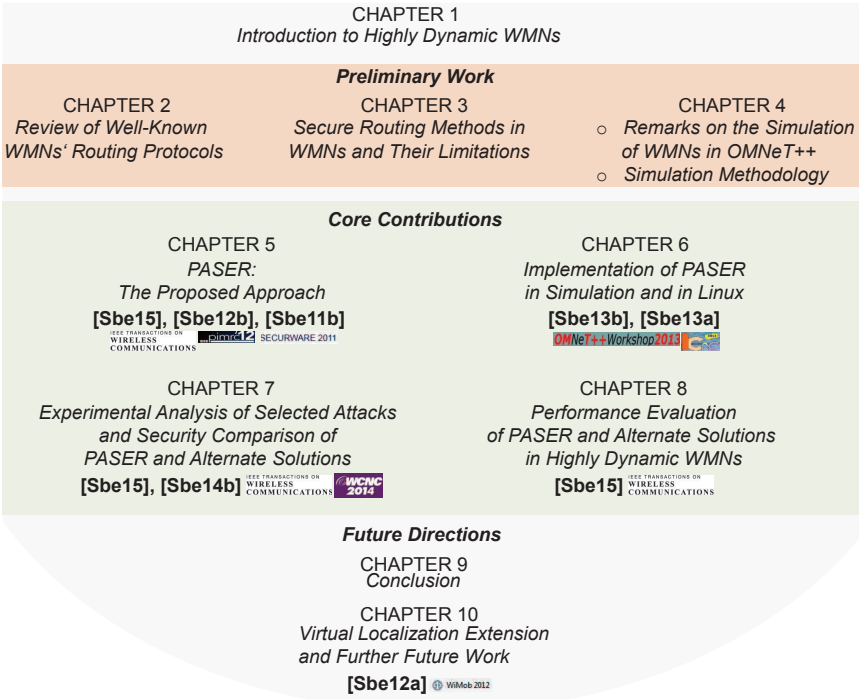

Figure 1.4: Overview of the organization of this thesis and the core publications of this research.

reviews existing secure routing research proposals and elaborates their deployment impediments in highly dynamic WMNs. The chapter closes with an outline of the improvements of the proposed approach.

- Chapter 4 describes the simulation of WMNs in the discrete event-based network simulator OMNeT++ and its INETMANET framework, including some remarks on the credibility of the results with respect to practice. In addition, the simulation methodology applied in this research to draw valid conclusions is presented.

- Chapter 5 gives an account of the novel proposed solution approach, PASER. The assumptions, secure routing goals, and building blocks of PASER are described. The costs of the PASER security operations are presented.

- Chapter 7 covers the modular implementation design of PASER in OMNeT++, the cross-layer implementation of PASER in Linux, and selected feasibility demonstrations of PASER.
- Chapter 8 explores, based on experimental tests, the vulnerability of the IEEE 802.11s/i security mechanisms against WMNs routing attacks. Besides, a comprehensive theoretical security comparison of PASER, ARAN, HWMPS, and BATMANS is given. This chapter emphasizes the necessity of a secure routing protocol combined with a dynamic key management scheme, such as is provided by PASER, to establish secure WMNs.
- Chapter 9 deals with the performance evaluation of PASER and its counterparts (ARAN, HWMPS and BATMANS) in two representative scenarios for highly dynamic WMNs: (1) on-demand ubiquitous network access and (2) efficient exploration of sizable areas in disaster relief using UAV-WMN. Variable network sizes and node densities are investigated based on the network simulator OMNeT++, realistic mobility patterns of UAVs, and an experimentally derived channel model. An analysis of the route discovery delay and the message overhead is also provided in this chapter.

The remaining chapters are a recapitulation of the main conclusions of this thesis and directions for new avenues to explore.

2

Review of Routing in Highly Dynamic WLAN Mesh Networks

Contents of this Chapter

In this chapter, the three routing philosophy classes of wireless multi-hop networks are explored, namely, proactive, reactive, and hybrid. The most-established routing protocols of each class are described, and these are discussed in light of the characteristics of highly dynamic WMNs.

2.1 Introduction to Routing in Highly Dynamic WMNs

The routing in highly dynamic WMNs is a critical building block that strongly influences the performance and reliability of the network. Although the adaptation of approaches used in wired networks has guided the first steps for routing in wireless multi-hop networks, routing protocols designed and applied in wired networks are not suitable in those networks, see [Adj03, Bak03, Wol94]. Due to the unique characteristics of wireless multi-hop networks in general and of highly dynamic WMNs in particular, there is a need for novel routing solutions. Table 2.1 illustrates the main differences between highly dynamic WMNs and wired networks with respect to routing. In contrast to interference-free wired networks, where nodes are mainly static, links are symmetric, and the throughput per link is fixed, highly dynamic WMNs are envisioned to have the following properties:

Table 2.1: Main routing differences between highly dynamic WMNs and wired networks.

Highly dynamic WMNs	**Wired networks**
Frequent connectivity changes (mobile nodes, wireless propagation phenomena)	Infrequent connectivity changes (static nodes, wired links)
Interference-prone links	Interference-free links
Symmetric and asymmetric links	Symmetric links
Variable link throughput	Fixed link throughput
Broadcast/multicast at low PHY data rates	Broadcast/multicast at high PHY data rates

- Highly dynamic WMNs have rapidly-changing connectivity due to the mobility of the nodes and due to radio propagation dynamics (e.g., fading). As a result, node pairs are intermittently connected and network links often break.

- Links in (highly dynamic) WMNs might be asymmetric, i.e., if a node A successfully receives a routing message from node B, this does not imply that node B can successfully receive the routing messages of node A.

- The throughput per link in (highly dynamic) WMNs varies depending on the received signal strength, which is subject to radio propagation dynamics and changes in the environment.

- In (highly dynamic) WMNs, routing messages are prone to collisions (due to interference) and channel errors. That is, the successful reception of routing messages is not guaranteed.

- Broadcast routing messages in (highly dynamic) WMNs are sent using low PHY data rates, because according to IEEE 802.11 (see clause 9.7.5.3 in [IEE11]), broadcast frames must be transmitted using a basic rate at which any receiver must be able to decode them (Differential Binary Phase Shift Keying (DBPSK) + Direct Sequence Spread Spectrum (DSSS) is generally used in practice to support legacy IEEE 802.11 devices —1 Mbit/s). This specification also increases the robustness of broadcast frames against channel errors as IEEE 802.11 does not implement any mechanism that enables the sender to detect whether the broadcast frames are successfully received. This requirement makes broadcast routing messages capacity-consuming in the shared medium of (highly dynamic) WMNs. Moreover,

because the transmission ranges of the broadcast frames are likely to overlap, classical broadcasting by flooding can lead to a broadcast storm due to redundancy, contention, and collision [Ni99].

Given these facts, a routing protocol in a highly dynamic WMN needs to maintain a consistent and stable network topology. It should dynamically adapt to variable conditions in the network, and it should feature relatively low routing overhead.

There are mainly two possibilities to perform routing in highly dynamic WMNs:

1. Using or adapting routing algorithms derived for MANETs as highly dynamic WMNs have many features in common with these networks. For instance, the characteristics listed in Table 2.1 also hold in MANETs.
2. Designing new routing protocols specifically tailored for highly dynamic WMNs (or adapting existing WMN protocols), which exploit the unique properties of these networks (see Table 1.1) and take into consideration the mobility of the backbone nodes.

In this regard, the different implementation designs and philosophies of the routing protocols in MANET and WMNs, that are relevant for highly dynamic WMNs, are covered and discussed next.

2.2 Routing Implementation Designs in Highly Dynamic WMNs

There are mainly two approaches to implement routing protocols in highly dynamic WMNs, namely, Internet Protocol (IP)-based or Media Access Control (MAC)-based. In the former approach, IP addresses are used to route data packets, and UDP routing messages are typically exchanged to establish the routes. The latter approach is based on MAC addresses, and MAC frames are utilized to build the routes.
Using appropriate IP addressing schemes, IP-based routing allows setting up hierarchical networks, to divide the network into clusters, and to differentiate between locally and globally relevant routing information, thus increasing the scalability of the network. In contrast, MAC-based routing is flat, and so scalability poses a big challenge. Thereby, solutions that use MAC-based routing such as the IEEE 802.11s mesh standard mainly addresses small and medium networks (up to 32 forwarding nodes) [Aok06]. IP-based routing protocols also feature easier implementation and a more portable code than MAC-based routing protocols. The routing logic in the former case can be implemented in user space. In the latter case, all protocol functionalities reside in the kernel space, which requires a deep understanding of kernel internals, and this makes the protocol implementation kernel-dependent.

On the flip side, IP-based routing cannot catch variations in link conditions without relying on indirect measurements provided by the MAC layer [Per13], which requires the definition of an interface between both layers that is difficult to standardize or realize in existing products. Using MAC-based routing, the information provided by the MAC layer and to some extent by the physical layer can be used to improve routing decisions. Besides, the common functionalities (synergies) of routing and MAC protocols can be exploited (e.g., using beacons for link sensing). Moreover, MAC-based routing is not bound to a specific network layer addressing scheme. Thus, a migration from IPv4 to IPv6 would be straightforward. In addition, using MAC-based routing, the WMN forms a single broadcast domain supporting a transparent delivery of broadcast data frames to all nodes in the network.
Examples of IP-based routing in highly dynamic WMNs include all protocols designed by the IETF MANET group [IETb], such as the Optimized Link State Routing (OLSR) [Cla03] and Ad Hoc On Demand Distance Vector protocol (AODV) [Per13]. A prominent example of MAC-based routing is the HWMP protocol specified in the mesh standard IEEE 802.11s [IEE11].

2.3 Routing Philosophy Classes in Highly Dynamic WMNs

Classification criteria of routing protocols in highly dynamic WMNs include, but are not limited to, routing philosophy, network organization (i.e., hierarchical or flat), location awareness, and mobility management, see [Wah06]. According to the first criterion, routing philosophy, WMNs' routing protocols can be divided intro three classes as illustrated in Figure 2.1, namely, proactive, reactive, and hybrid. In the case of proactive protocols, the nodes periodically send routing messages throughout the entire network, regardless of the network traffic, and each node has the routing information of all other reachable nodes in the network. In contrast, reactive protocols are traffic-aware and their overhead is strongly related to the traffic load in the network. Nodes only exchange routing information when a route to an unknown destination is required and only information about active routes is maintained. Hybrid protocols are a combination of both previous routing classes. Typically, only nodes having a superior role, e.g., gateways, periodically broadcast messages throughout the network. A profusion of routing protocols of each class has been proposed in the last decade [Abo04]. In this section, the different routing philosophies and well-established protocols corresponding to each philosophy are presented in detail, and these are discussed in light of the characteristics of highly dynamic WMNs.

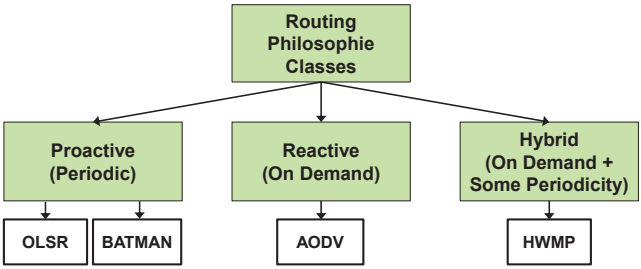

Figure 2.1: Routing philosophy classes and example protocols in WMNs.

2.3.1 Proactive Routing

Proactive routing protocols maintain routing tables that include information about all reachable nodes in the network. Every node propagates routing information throughout the network at periodic time intervals (and/or conditionally, if a change in the topology occurs). The difference between routing protocols of this class exist in the strategies used to update the routing tables, the number of tables maintained, and the type of information kept in the tables [Abo04]. Using shortest path algorithms such as Dijkstra, a node can find the best route to the destination with negligible delay, as it has a complete knowledge of the network topology, thus, the route to any reachable destination is always available. This class of protocols has, however, a large control overhead due to the periodic exchange of routing information. Besides, there is a convergence time whenever a change in the topology occurs or when the network is first set up. This is the time needed to synchronize the tables of all nodes. OLSR [Cla03] and BATMAN [BATa] belong to the best known protocols of this class.

Proactive protocols are envisioned to be suitable for less dynamic mesh backbones with sporadic traffic, high traffic load, and long routes between communication pairs [Xu10, Huh04]. The scalability of this routing class is constrained by the size of the routing tables and the overhead of the routing update messages [Abo04].

2.3.1.1 Optimized Link State Routing (OLSR)

OLSR [Cla03] is an IETF experimental IP-based proactive routing protocol tailored for wireless multi-hop networks. It received attention in community WLAN mesh networks such as Freifunk Berlin [FRE], Guifi.net [GUI] and the Athens wireless metropolitan network [AWM]. It is an optimization of classical link state

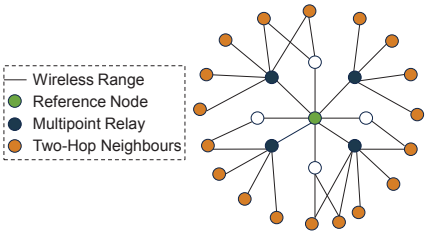

Figure 2.2: Example of the multipoint relay optimization in OLSR.

routing protocols [Per99]. In a link state routing protocol, nodes typically take the following steps:

1. Detect all local connections (also known as link sensing or neighbor discovery);
2. Flood the network with a link state message that includes, among others, information about all links (i.e., direct neighbors and connection metrics);
3. Build a map of the network with all received advertisements;
4. Calculate the shortest path to destinations, e.g., using Dijkstra.

In OLSR, this approach is optimized by introducing a novel concept termed MultiPoint Relays (MPRs). Rather than advertising all links of all nodes, only MPRs generate link state messages, and they only declare the set of neighbor-links that have selected them as MPRs. Besides, only MPRs forward link state messages. As result, the size of the routing messages is squeezed, and the number of broadcast retransmissions is reduced.

MultiPoint Relay (MPR) Selection Approach. The MPRs of a node are defined as the minimal subset of one-hop neighbors, the union of which provides links to all two-hop neighbors, see Figure 2.2. To select its MPR, a node uses mainly hello messages and takes the following steps:

- It exchanges periodically hello messages with one-hop neighbors to sense direct links;
- It includes a list of one-hop neighbors in its hello messages. In this way, the information about the two-hop neighbors is exchanged, thereby, each node gets knowledge about its one and two-hop neighbor sets;

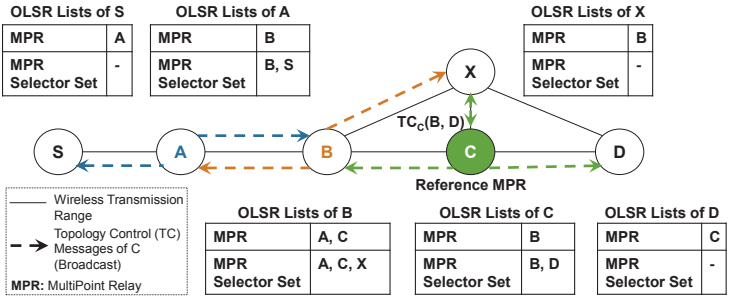

Figure 2.3: Example of MPR, MPR selector sets, and the retransmission of topology control messages in OLSR.

- It selects MPRs which cover all its two-hop neighbors and signals this set via hello messages to its one-hop neighbors, see an example of MPRs in Figure 2.2.
  In this example, four broadcasts are transmitted in case of OLSR to diffuse the information about reference node to its two-hop neighbors. In the case of a classical link state routing protocol, nine broadcast messages were needed: one broadcast message of the reference node and 8 *retransmissions by the one-hop neighbors.*

Link State Advertisements. To advertise link states, Topology Control (TC) messages issued by MPRs are disseminated in the network. An MPR generates TC messages advertising only those nodes in its MPR selector list. This list comprises the one-hop neighbors that have selected the node as their MPR. All nodes in the network process the TC messages, but a node only forwards TC messages if the sender is in its MPR selector set. Figure 2.3 illustrates an example of a network setup and the corresponding MPRs and MPR selector lists. In this example, only A, B and C are MPRs, thus, only these generate TC messages. When C generates a TC message, it reports only about its links with B and D, its MPR selectors. Only B retransmits this message, as only B has C in its MPR selector set. Analogously, A and C should retransmit the TC received from B. However, as C is the originator of this message, only A forwards it. As Figure 2.3 shows, the TC message of C reaches all nodes in the network, S will know about D from this message and about X from the TC messages of B. That is, due to the fact that the MPRs are selected in such a way that all nodes in the network are covered, the routing information is distributed throughout the whole network despite having limited the number of retransmissions and reduced the amount of information included in the routing messages.

Table 2.2: Performance-crucial parameters of OLSR.

Parameter	Description & refresh conditions
Hello-Interval [s]	Hello messages are necessary to establish and refresh links between one- and two-hop neighbors.
TC-Interval [s]	TC messages are necessary to establish and refresh routes between nodes.
Neighbor-Hold-Time [s]	When this timeout is triggered, the corresponding neighbor entry is set as invalid (or deleted). All the route entries for which this neighbor has been next-hop are also set as invalid (or deleted). The timer is refreshed only in case of receiving routing messages from the corresponding neighbor.
Route-Hold-Time [s]	When this timeout is triggered, the corresponding route entry is set as invalid (or deleted) in the routing table. The timer is refreshed only in case of receiving corresponding TC messages.
LinkQualityLevel	This sets the level to which ETX is used. 0 disables ETX. 1 enables ETX for MPR selection. 2 uses ETX to select MPR and calculate the routing table.

The most performance-crucial parameters of OLSR are described in Table 2.2.

Routing Metric. To find the best route to a destination, OSLR uses a shortest path algorithm applied to a weighted graph of the network topology. To weight the edges of this graph (i.e., the network links), OLSR uses the Expected Transmission Count (ETX) metric [DeC05]. The ETX of a link estimates the number of transmissions (including retransmissions) required to successfully deliver a packet over that link. The ETX of a route is the sum of the ETXs on all links. The route with the lowest ETX is chosen as the best route, as it offers the highest goodput (i.e., received bits per second at the destination).
According to IEEE 802.11, a unicast data frame is successfully sent over a link if it is successfully received by the one-hop neighbor and the sender gets an acknowledgment as a response. That is, a retransmission only occurs if one these two conditions is violated. Let each node sends a probe message every α [s] seconds (e.g., hello messages every hello-Interval). Let β [s] be the period of seconds to calculate ETX with $\beta > \alpha$. Let the number of successfully received

probe messages during β seconds be n_β. The delivery ratio of sending a probe message can be calculated as $n_\beta / \frac{\beta}{\alpha}$. Let the Link Quality (LQ) be the delivery ratio of sending a probe message from the one-hop neighbor to the probing node, and the Neighbour Link Quality (NLQ) the quality of the link in the opposite direction. The probability for a successful packet transmission is then $(NLQ \cdot LQ)$. Thus, ETX can be calculated according to Equation 2.1.

$$ETX = \frac{1}{NLQ \cdot LQ} \tag{2.1}$$

To enable the calculation of the ETX of a link, hello messages are extended to include the NLQ values of the one-hop neighbors. To calculate the ETX of a route, the required knowledge of the ETXs of non-direct links that build the route is acquired through information disseminated in TC messages.
While the ETX captures the packet loss in both directions of a link, and detects interference among the links of the same route, ETX probe messages have a fixed size and they experience different packet loss ratios than unicast data packets, because broadcast messages use more robust modulation and coding schemes, and thus have low transmission rates. That is, the loss rate of broadcast messages may not be the same as that of the data packets.

While the MPR concept has been proven to optimize link state routing in theory, the deployment of OLSR in community networks has led to miserable performance. Nodes have selected non-reliable links to build a minimal set of MPR. Due to link impairments in WMNs [Mil07], this resulted in inconsistent routing tables and, thereby, routing loops occurred. It was also observed that routing tables broke in only a short time (e.g., due to interference) but took a long time to build (the need to sense neighbors, to select MPRs, afterwards, topology control messages are first advertised). For more details about the limitations of OLSR in practice, see the OLSR.org story and the references therein [OLS]. To address these limitations, the BATMAN protocol was proposed [BATa, Neu08].

2.3.1.2 Better Approach To Mobile Ad-hoc Networking (BATMAN)

The BATMAN routing protocol was developed to address the problems associated with OLSR in practice [Joh08, Bar09]. It is a proactive WMNs' routing protocol, by which nodes do not require a complete knowledge of the network as they do in OLSR (or other proactive link state protocols). In contrast, the nodes are only concerned to determine the best next-hop towards each destination. In this way, the inefficiency inherited in link-state algorithms in WMNs due to stale topology-graphs and inconsistent routing tables is avoided.
To find the best next-hop towards a destination, BATMAN nodes periodically

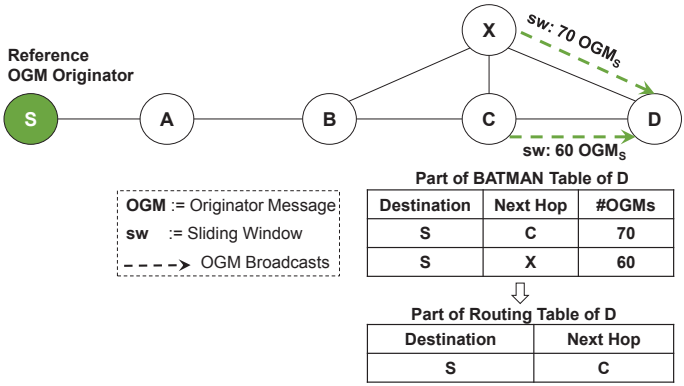

Figure 2.4: Example of finding the bext next-hop towards a destination in BATMAN.

flood the network with OriGinator messages (OGMs). On the one hand, they announce their existence to the network. On the other hand, the reception or loss of these messages is used to indicate the quality of the routes: the best one-hop neighbor towards a destination is the next-hop that forwards the most of that destination's OGMs within a sliding window. Thus, lost OGMs due to link impairments do not lead to inconsistencies in the network, they are rather used for better routing decisions. OGMs are relatively small messages (52 Bytes). They contain a limited amount of information:

- IP addresses of originator and of forwarder
- A time to live value (to constraint flooding)
- Sequence number (to avoid reprocessing of an OGM that has been already received).

OGMs sequence numbers are also a key piece of information for selecting the best next-hop towards a destination. BATMAN uses a sliding window to record the most recently received sequence numbers. The amount of sequence numbers recorded in the sliding window is used as a metric for the route quality. Figure 2.4 illustrates an example by which node D receives within a sliding window 70 OGMs of S via node X and 60 OGMs of S via node C. As a result, D chooses node C as the best next-hop towards S.
The most crucial parameters of BATMAN with respect to performance are described in Table 2.3.
The BATMAN approach is shown to be reliable and loop-free in practice [Abo09,

Table 2.3: Performance-crucial parameters of BATMAN.

Parameter	Description & refresh conditions
OGM-Interval [s]	OriGinator Messages (OGM) are necessary to establish and refresh routes between nodes.
Route-Hold-Time [s]	When this timeout is triggered, the corresponding route entry is set as invalid (or deleted). Timer is refreshed only in case of receiving the corresponding routing message.

Mur10], and it offers better scalability than OLSR [Joh08, Bar09]. The author of this thesis shows in [Sbe14c, Poj11] that BATMAN performs better than OLSR in selected highly dynamic WMNs. Currently, BATMAN is deployed in different community networks, which include Freifunk Berlin [FRE], Guifi.net [GUI] and the Athens wireless metropolitan network [AWM]. Noteworthy with respect to the implementation of BATMAN are the different branches [BATb] that have evolved over time, such as BATMAN-adv (MAC-based) and BMX6 (IPv6-based), with BATMAN referring to the IPv4-based implementation. In the rest of this thesis, BATMAN is used as a representative of proactive WMNs' routing protocols.

2.3.2 Reactive Routing

Reactive routing protocols attempt to reduce the control overhead associated with proactive routing at the expense of a delay in finding routes. They operate on-demand, i.e., routes are only looked up when needed (e.g., when a node requires to send data to an unknown destination), and they maintain only active routes, i.e., a node only updates information about routes currently in use (thus, the routing tables are smaller than in proactive protocols). To find routes, a route discovery process composed of two steps is taken.

1. Flooding a Route REQuest (RREQ) issued by the source in the network. In the meantime, the data packets get buffered by the source.

2. Sending back a Route REPly (RREP) issued by the destination or by an intermediate node that has a route to the destination. The reverse route is used to send back the RREP using bidirectional links. In the presence of a unidirectional link, mechanisms such as BlackListing, hello messages, and ReversePathSearch are used, see [Mar02].

A well-known reactive routing approach is the AODV protocol [Per13].

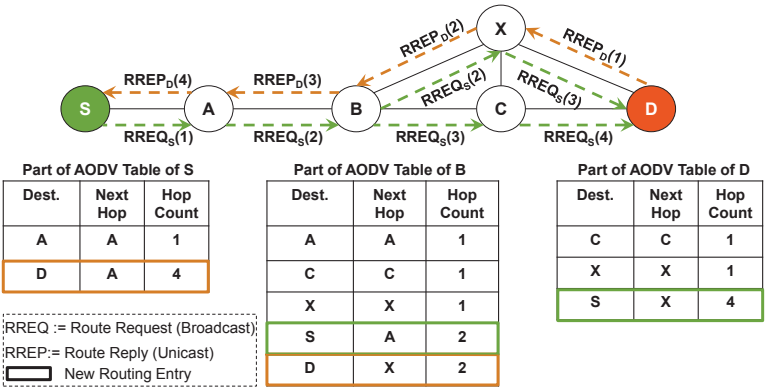

Figure 2.5: Example of the route discovery in AODV.

Reactive protocols are expected to perform better than proactive protocols in highly dynamic mesh backbones with static traffic and moderate route lengths between communication pairs [Sbe14c, Poj11, Lew10, Hsu04, Joh99], see also [AR12] and the references therein. The scalability of this routing class is mainly constrained by the flooding overhead of route discoveries [Abo04] and the number of route breaks (i.e., traffic load and route length) [Vie04].

2.3.2.1 Ad Hoc On Demand Distance Vector protocol (AODV)

AODV [Per13] is an IETF experimental IP-based reactive routing protocol. It builds the basis for the routing framework of the IEEE 802.11s mesh standard [IEE11]. It uses periodic hello messages to sense neighbor-links. It uses the classical reactive request–reply mechanism to discover routes. The distance (number of hops) to reach a destination, the hop count, is used as a route metric. Figure 2.5 illustrates an example of the route discovery in AODV. Hereby, S starts a route discovery towards D. As the figure shows, the intermediate nodes always store the reverse paths towards the nodes that triggered the route discovery, and they always increment the hop count by one when forwarding the route discovery messages. For instance, node B creates a route entry for S upon processing its route request. This route entry is used to send back the route reply. B increments the hop count before retransmitting the message. AODV also supports intermediate replies, i.e., intermediate nodes can reply on behalf of the destination if they already know the route to that destination. After establishing the route, the data packets are forwarded hop-by-hop towards the destination. That

Table 2.4: Performance-crucial parameters of AODV.

Parameter	Description & refresh conditions
Hello-Interval [s]	Hello messages are necessary to establish and refresh links between one-hop neighbors.
Neighbor-Hold-Time [s]	When this timeout is triggered, the corresponding neighbor entry is set as invalid (or deleted). All the route entries for which this neighbor has been next-hop are also set as invalid (or deleted). Timer is refreshed upon receiving a hello message of the one-hop neighbor.
Route-Hold-Time [s]	When this timeout is triggered, the corresponding route entry is set as invalid (or deleted). Timer is refreshed every time a node sends or receives an IP-packet over the route.

is, each intermediate node itself decides, based on its routing information, how to optimally forward the data packet.
The most crucial parameters of AODV with respect to performance are described in Table 2.4.

To detect broken links, a necessity for route maintenance, AODV uses two mechanisms, as illustrated in Figure 2.6 (top).

1. It uses periodic hello messages between one-hop neighbors. If a node does not receive a hello menage from its one-hop neighbor within a given interval, typically 3·hello-Interval, the link is considered broken.

2. It uses a link layer feedback mechanism to enable a fast reaction to route breaks in the case of active data transfer. If a node does not receive an acknowledgment for an unicast frame sent to the next-hop towards a destination, even after seven retransmissions, which is the default number for data-frame retransmissions according to IEEE 802.11, the link is considered broken.

After detecting a broken link, a forwarding node broadcasts a Route ERRor (RERR) message in the network. This message includes information about the unreachable next-hop as well as a list of all the nodes for which the unreachable node was the next hop, if available. When a node receives a RERR message, it checks whether the sender of the RERR is the next-hop to the unreachable node(s). In that case, the route is marked as invalid and the node rebroadcasts

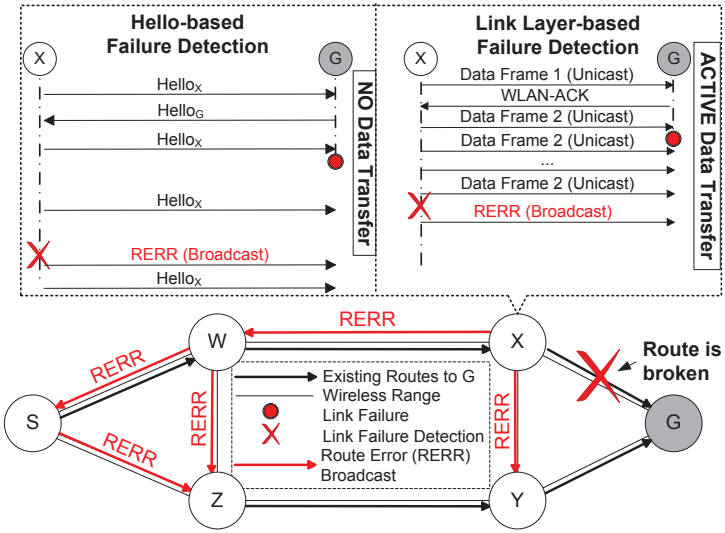

Figure 2.6: Detection of route breaks in AODV.

the route error message. Figure 2.6 (bottom) shows an example in which a node X loses the route to the gateway. As this figure shows, X generates a RERR message, and only W and S rebroadcast this message, since only those nodes have X and W respectively as next hops towards G. This RERR propagation mechanism enables more efficient network topology awareness than does a simple flooding. Note that AODV also supports local route repair, i.e., after detecting the link failure, X might try to rediscover the route towards G before generating the RERR (e.g., if the distance between X and the destination is smaller than that between X and the source).

To avoid routing loops and a consequent counting to infinity problem in the face of anomalous delivery of routing messages, AODV uses destination sequence numbers to make sure that only the latest information is considered. Hereby, a destination sequence number is maintained for each destination in the AODV routing table. This number is a monotonically increasing number, and the originating nodes include it in each routing message.

2.3.3 Hybrid Routing

This class of routing combines reactive and proactive routing protocols to exploit their advantages and mitigate their limitations. There are mainly two philosophies of hybrid routing:

1. Dividing the network into regions or zones. Applying proactive routing within each zone and reactive routing between zones. This should reduce the route discovery overhead between nodes in close proximity.
2. Running reactive routing on all nodes and additionally a proactive tree routing on special nodes, e.g., gateway or common destinations. This should minimize the route discovery delay and the number of route discoveries.

In this thesis, the focus lies on the second philosophy, as it is used in the standardized mesh routing protocol HWMP [IEE11].

Hybrid protocols are anticipated to be more scalable and suitable for a wider variety of mesh networks than reactive and proactive protocols [IEE11].

2.3.3.1 Hybrid Wireless Mesh Protocol (HWMP)

In this sub-section, an overview of the key features of the IEEE 802.11s mesh standard [IEE11] is given. Then, the functionality of HWMP, which is the default routing protocol specified in the IEEE 802.11s standard, is elaborated.

Key Features of IEEE 802.11s. This paragraph summarizes the key features of the IEEE 802.11s mesh standard. The reader is referred to [Hie10, Hie11, IEE11] for detailed information.

- **Interworking with IEEE 802 Networks**: The mesh standard specifies a set of mechanisms for interoperability with IEEE 802 networks: A mesh network appears as a single Ethernet segment to external networks. It implements a single broadcast domain supporting a transparent delivery of uni-, multi-, and broadcast frames to destinations inside and outside the mesh network. To inform the mesh backbone nodes about the existence of external networks, the mesh gateway nodes periodically broadcast the appropriate announcements.
- **Mesh Self-Formation**: To discover peers in proximity (one-hop neighbors), mesh nodes perform a passive scanning (by observing periodic beacons) or an active scanning (using probe messages). Candidate neighbors are identified based on new information elements included in beacons and probe response frames (e.g., mesh ID and mesh configuration element).

Once such a candidate peer is identified, a mesh node applies a mesh peering protocol to establish a link with that neighbor.

- **Medium Access Control (MAC) Enhancements:** The IEEE 802.11s defines a mesh coordination function for medium access control. The mesh coordination function specifies a mandatory and an optional scheme. The mandatory scheme is based on the Enhanced Distributed Channel Access (EDCA) of the IEEE 802.11e standard. This scheme enhances the legacy IEEE 802.11 Distributed Coordination Function (DCF) to support Quality Of Service (QoS) by introducing priorities into the data transmitted over the network. Note however that the EDCA does not guarantee QoS, it only makes it more probable. The optional scheme is referred to as Mesh Coordination Controlled Channel Access (MCCA). It is a contention-free distributed channel access reservation protocol. Nodes reserve transmission time in the future to avoid frame collisions. Hereby, beacon frames are used to disseminate the reserved transmission time of a node to its two-hop neighbors. However, this scheme requires a tight synchronization of the mesh nodes.

- **Generic IEEE 802.11 Physical Interface:** The IEEE 802.11s supports multiple channels and multiple radios. It might run on top of any IEEE 802.11 physical interface, which includes, but is not limited to, IEEE 802.11 a/b/g/n/ac/ax.

- **Security of Peers**: With IEEE 802.11s, nodes do not have static security roles, e.g., supplicant or authenticator as in legacy IEEE 802.11 networks where the IEEE 802.11i security mechanisms are applied. In a mesh network, a mesh node is a supplicant when it joins the network, and it is an authenticator when a new neighbor-node joins the network. Thereby, in IEEE 802.11s, the security architecture of IEEE 802.11i has been extended to meet the specifics of WMN. The heart of the IEEE 802.11s security architecture is the dictionary attack-resilient Simultaneous Authentication of Equals (SAE) algorithm.

- **Path Selection and Radio-Aware Metric**: Routing is called path selection in IEEE 802.11s, as this mechanism is implemented at the MAC layer. In this regard, IEEE 802.11s specifies an extensible path selection framework. This framework includes multiple path selection protocols and metrics for flexibility, and new proposals could be developed. A mandatory protocol is HWMP and a mandatory metric is the airtime link metric.

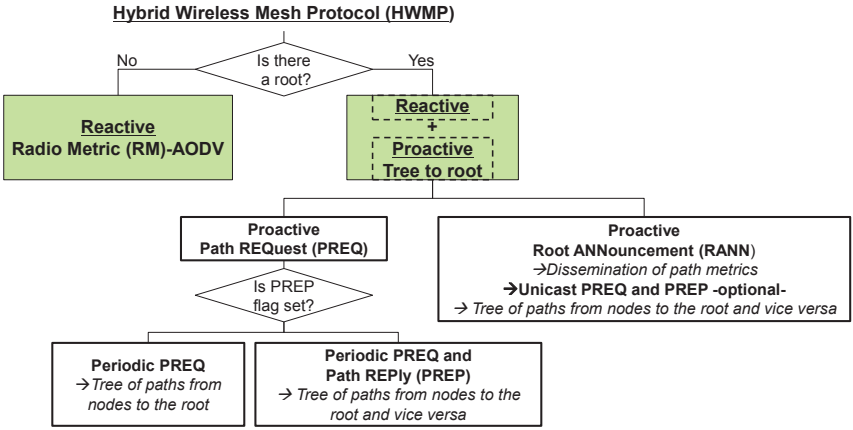

Figure 2.7: HWMP modes of operation.

Functionality of HWMP. While the IEEE 802.11s standard supports the use of other routing protocols, it specifies HWMP as mandatory for compatibility purposes. In contrast to traditional routing schemes which are IP-based, HWMP operates at the MAC layer and uses MAC addresses to forward packets. Hence, it is called a 'path selection' algorithm instead of a 'routing' algorithm. It is a hybrid path selection protocol that combines the flexibility of on-demand path discovery in highly dynamic networks and the efficiency of proactive path selection for low mobility mesh gateways. The different modes of operation of HWMP, depending on its configuration, are illustrated in Figure 2.7.
Reactive path selection is a mandatory functionality in HWMP. It is based an a Radio Metric (RM)-AODV, which, by default, uses a metric called airtime link for path selection, instead of using the hop count. The goal of this metric is to save network capacity by estimating the time consumed to transmit a test frame, taking into account the PHY data rate at which the frame is transmitted, the overhead posed by the PHY implementation in use, and also the probability of retransmission, which is related to the link error rate. Consequently, the path having the minimum airtime cost is the best. The airtime link metric, C_a, is defined in Equation 2.2. The parameters used in this equation are described in Table 2.5.

$$C_a = \left[O + \frac{B_t}{r}\right] \frac{1}{1 - e_f} \tag{2.2}$$

Table 2.5: Airtime link metric parameters.

Parameter	Description
O [s]	Constant channel access overhead (latency) that depends on the PHY layer configurations.
B_t [bit]	Length of the test frame (constant).
r [Mbit/s]	Transmission data rate for sending B_t based on current conditions. Its estimation is dependent on local implementation of rate adaptation mechanisms.
e_f	Test frame error rate for B_t. Its estimation is a local implementation choice [Car11]. In the open802.11s implementation [OPE], each node maintain a *fail-avg.* parameter for each link, a moving percentage of failed unicast frames. This information is mainly used to calculate e_f.

Proactive path selection of HWMP is an extension, which is enabled when a node is configured as a root element, as illustrated in Figure 2.7. A root element supports the following two exclusive modes of operation (see [IEE11, Bah07] for more details):

1. *Proactive PREQ mechanism:* This mode uses the proactive Path REQuest mechanism (PREQ), i.e., the root element periodically propagates a proactive PREQ in the network. The processing of the proactive PREQ depends on the Path RESponse (PREP) flag field as follows.

   - Without PREP Flag: If the PREP flag is not set, each node records the metric and hop count to the root in its HWMP routing table, updates these fields in the PREQ, and retransmits the message. In this way, the presence of a root node and information about the paths towards this node is disseminated in the whole network. No PREP is sent in response to the reception of a proactive PREQ if the PREP flag is not set.

   - With PREP Flag: If the PREP flag is set, nodes must send a PREP in response to the reception of a proactive PREQ. In this way, all nodes maintain the path to the root node and the latter keeps track of paths to all nodes. As a benefit, in case a source wants to send data to a destination but has no path to that destination, the path through the root node can be used until a more efficient path is discovered by the reactive part of HWMP.

2. *Proactive RANN mechanism:* In this mode, the path information (e.g., metric and distance) towards the root node are maintained by periodically

Table 2.6: Performance-crucial parameters of HWMP.

Parameter	Description & refresh conditions
RootMode	A value of 0 indicates that the node is not a root element. A value of 1 activates the RANN announcements. A value of 2 corresponds to proactive PREQ without PREP flag. A value of 3 means proactive PREQ with PREP flag. If applicable, RANN-Interval and PREQ-Interval are relevant parameters that determine the interval of sending proactive frames.
Neighbor-Hold-Time [s]	When this timeout is triggered, the corresponding neighbor entry is set as invalid (or deleted). All the route entries for which this neighbor has been next-hop are also set as invalid (or deleted). Timer is refreshed upon sending or receiving a frame to or from the neighbor.
Route-Hold-Time [s]	When this timeout is triggered, the corresponding route entry is set as invalid (or deleted). Timer is refreshed every time a node sends or receives a frame over the route.

broadcasting Root ANNouncement (RANN) messages by this node into the network. The RANN mechanism does not establish a path. If a node wants to create a path to the root node, unicast PREQ-RREP are sent.

The most crucial parameters of HWMP with respect to performance are described in Table 2.6.

Having the specifics of highly dynamic WMNs in mind, the question that arises when aiming to design an efficient secure routing protocol: What routing philosophy class is mostly suitable in highly dynamic WMNs? As aforementioned in this chapter, it is shown in [Sbe14c, Poj11, Xu10, Lew10, Hsu04, Joh99] that reactive protocols perform better than proactive protocols in case of highly dynamic backbone nodes, fixed source-destination pairs, and moderate route lengths. That is, reactive protocols are better suited for small scale UAV applications where most of the traffic is destined from the UAVs to the ground control station. Thereby, the secure routing protocol proposed in this thesis is based on the reactive routing philosophy. This philosophy might be easily modified in the future to implement a hybrid approach, if necessary.

3

Secure Routing Issues in Highly Dynamic WLAN Mesh Networks

Contents of this Chapter

In recent years, experimental research has focused on different IEEE 802.11 mesh network approaches to realize an UAV-WMN. For instance, in the UAVNet airborne network [Mor12], IEEE 802.11s is used to interconnect the aerial nodes. It is shown that such an UAV-WMN strongly increases the network capacity in comparison to a ground-WMN. Another implementation of UAV-IEEE 802.11s is analyzed in [Pat13], where the impact of the radio signal strength on the autoconfiguration of two UAVs and a ground station is investigated. The results prove that IEEE 802.11s is a promising WMN solution for collaborative UAV applications. In [JP12], a different approach is considered. A combination of IEEE 802.11a/g/n and IP-based routing protocols is used to establish a dynamic small scale aerial network. It is demonstrated that such a combination is also an option for successfully deploying an UAV-WMN. Yet, an open issue, which has not been thoroughly considered yet, is the security of UAV-WMN —it is supposed that the mechanisms deployed in WMN can be adopted.
The surveys in [Sgo13, Sen13, Nav08] present a comprehensive analysis of the security in WMNs. They point out that several attacks are common in wireless networks such as jamming at the PHY layer, and these can be mitigated by conventional security mechanisms, while some attacks are specific to WMNs. The latter mainly includes attacks on the core service of the mesh backbone (i.e., routing), such as the blackhole and wormhole attacks, and user-related attacks, such as attacks on the user privacy with respect to data content, traffic flows,

and location. In this research, the focus lies on the security of the routing functionality in the UAV-WMN mesh backbone. For privacy preservation and other user-related security services in WMNs, several approaches have been proposed in [Lin12, Ren10, Wu08, Wu06, Zha06], which can be applied in combination with secure routing, if necessary. For instance, in disaster scenarios, end-to-end security mechanisms are already used to ensure the privacy of the data of rescue fighters [Bal14, Šub10], while the privacy of their traffic flows (source, destination) and their location are not really a concern as these information are predefined in their public regulations.
To secure the routing process in the backbone of (UAV-)WMN, two methods mainly exist:

Method 1: Combining well-established, non-secure routing protocols with standardized security mechanisms (for confidentiality and integrity), such as those of the IEEE 802.11i security standard [IEE04] or the IEEE 802.11s mesh standard [IEE11]. This method is applied as a contemporary solution in many commercial mesh routers, such as [FIR, TRO, RAJ, MES].
The limitations of this method with respect to routing attacks are described in Section 3.1, using the standardized security mechanisms of IEEE 802.11s/i as a reference.

Method 2: Use of a secure mesh routing protocol. This method is striven for in practice. Many secure routing protocols have been proposed in the last decade [Sgo13, Sen13, Abu08], yet, none of them has been deployed. The high overhead of the security mechanisms of these protocols or the strong

Table 3.1: Main characteristics of symmetric-key and public-key cryptographic algorithms.

Characteristic	**Symmetric-key**	**Public-key**
Key type	One secret key (or several secret keys easily derived from each other)	A pair of keys, one secret and one public, which cannot be derived from each other
Key length	Relatively short (e.g., 128 bit AES)	Relatively long (e.g., $\geq$ 1024 bit RSA)
Main shortcomings	Key distribution	Computationally intensive
Main application goals	Mutual authentication, data confidentiality and integrity	Mutual authentication, non-repudiation, and key establishment

assumptions taken during their design (e.g., the existence of an efficient symmetric key management scheme) have rendered their deployment in real life applications infeasible.
A review of selected secure routing proposals is given in Section 3.2, and their deployment impediments are elaborated in more detail.

For the sake of completeness, a brief introduction to cryptography is provided in Appendix A. It covers the definition of the main security goals of cryptography as well as relevant cryptographic algorithms, which are used in the aforementioned methods as well as in the secure routing protocol proposed in this thesis. These algorithms are typically divided into two classes: symmetric-key and public-key algorithms. Table 3.1 depicts an overview of their main characteristics. For a rigorous presentation of cryptography, the reader is referred to [Paa09] and [Men96].

3.1 Limitations of the IEEE 802.11 Security Frameworks

The IEEE 802.11i security standard [IEE04] and the IEEE 802.11s mesh standard [IEE11] specify security frameworks that ensure secure links (one hop communication) between WMN nodes.
The IEEE 802.11i standard was released in 2004. The standard addresses the weaknesses of the Wired Equivalent Privacy (WEP) mechanism defined in the base standard, IEEE Std. 802.11 1999. Detailed information about these weaknesses are provided in [Edn03]. IEEE 802.11i specifies enhanced security services and mechanisms to resolve the loophole of WEP and to build a Robust Secure Network (RSN). One major enhancement in an RSN is to use the Counter Mode with the Cipher block Chaining Message authentication code Protocol (CCMP) to achieve the confidentiality and integrity of the frames. This protocol is based on the secure block cipher AES. An obstacle to using AES back then was that it was not backward compatible with the existing WEP hardware (e.g., the WLAN network interfaces), as AES requires the existence of a new, more powerful hardware engine. To allay industry concerns for already deployed systems, the Wi-Fi alliance (an alliance of major 802.11 vendors aiming to ensure product interoperability) took a subset of IEEE 802.11i and created the Wi-Fi Protected Access (WPA). WPA uses the Temporal Key Integrity Protocol (TKIP) for the confidentially and integrity of frames. This protocol does not require a hardware upgrade, as it is based on the same stream cipher as WEP, namely, Rivest Cipher 4 (RC4). While WPA solves the security problems of WEP, it has been found to be vulnerable to several attacks related to the RC4 cipher, see [Tew09]. This issue was resolved after the release of the IEEE 802.11i standard. The Wi-Fi alliance has adopted the standardized long term security solution based on CCMP, and it has termed this solution Wi-Fi Protected Access 2 (WPA2). Hereafter, the term

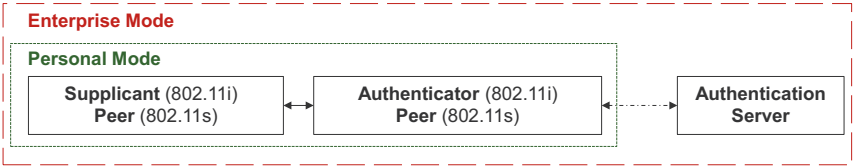

Figure 3.1: Modes of operation of the IEEE 802.11 security frameworks.

IEEE 802.11i is used to refer to WPA2.
The IEEE 802.11s standard was released in 2011. Its security framework is based on the IEEE 802.11i while taking the specifics of WMNs into consideration. For instance, in contrast to traditional WLANs, which typically consist of access points and clients, and where the access point includes an authenticator, and each associated client includes a supplicant, in the backbone of a WMN, there is no strict security hierarchy: all nodes are peers. For this reason, the authentication mechanism in IEEE 802.11s was changed to support peer roles. This and other relevant differences between IEEE 802.11i and IEEE 802.11s are elaborated throughout this section.

3.1.1 Security Goals and Modes of Operation

Implementing the IEEE 802.11 security frameworks mainly guarantee the following security goals:

- **Frame authentication**;
  - Authentication of nodes,
  - Integrity of frames,
- **Freshness of frames**;
- **Confidentiality of frames**.

The authentication and freshness of the frames prevent unauthorized network access and hinder an external attacker from injecting and modifying frames, or replaying frames at a later time in the network. Thus, an external attacker is disabled from mounting internal routing attacks or time-based replay attacks. The confidentiality of the frames is less relevant for the routing functionality [Egn10], it is rather necessary for data forwarding to prevent unauthorized nodes from eavesdropping on the data traffic.

The IEEE 802.11 security frameworks support two modes of operation, as illustrated in Figure 3.1. These are defined as follows:

- *Enterprise mode:* Three parties are always involved to establish a secure link: the two communicating parties and an authentication server. The authentication server is used to determine whether a node is authorized to access the network by examining the node's credentials (e.g., public key). This mode is not applicable in highly dynamic WMNs for the following reasons:

  – As this mode is designed for enterprise applications (i.e., static networks with typically a wired link to the authentication server), it has been shown in [Egn12, Zha08, Che06] that the enterprise mode is not suitable for dynamic and mobile WMNs, due to the high authentication delay values (multi-hop), and because there is no permanent reliable link to the authentication server in such environments. For instance, it is shown in [Egn12] that the authentication in this mode lasts for 650 ms in case of three hops. This does not satisfy the quality of user experience of multimedia streaming, where according to [Abo03] the delay should be below 150 ms.

  – In contrast to a WLAN with a secured wired link between the access point and the authentication server, in WMNs, a trustworthy wireless multi-hop route to the authentication server cannot be assumed [Wan08]. That is, there is an interdependency cycle of this mode with a secure routing protocol [Zha13b, Bob03].

  Apart from that, the enterprise mode uses the same protocol to ensure the confidentiality and integrity of the frames as the personal mode, and this protocol has security caveats with respect to routing in the mesh backbone, as shown in the next sub-section.

- *Personal mode:* This mode is based on a pre-shared key (password) to access the network. Here, no authentication server is required: the communicating parties can establish a secure link between each other without relying on a third party. Hence, this mode is applicable in highly dynamic WMNs. Thus, the focus lies on this mode in the rest of this thesis. Next, its security limitations in the backbone are elaborated.

3.1.2 Establishing Secure Link—Personal Mode

To establish a secure link in the personal mode, nodes go through the following four phases. These are depicted in Figure 3.2.

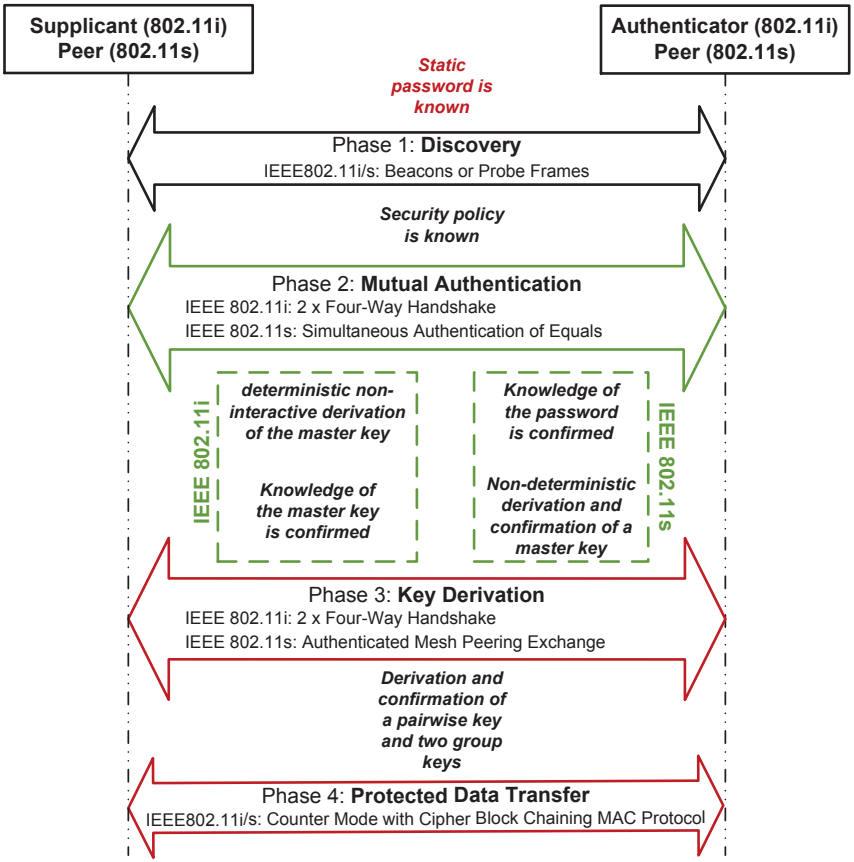

Figure 3.2: Steps to establish a secure link in the personal mode of the IEEE 802.11 security frameworks.

1. *Discovery*: In both security frameworks, IEEE 802.11s/i, each node advertises its security policy in probe frames (active scanning) or periodic beacons (passive scanning). Other nodes receiving these frames get to learn the security policy of the advertising node.

2. *Authentication*: According to IEEE 802.11i, a four-way handshake is launched twice to mutually authenticate two communicating parties in a WLAN multi-hop network. Here, each node performs once as authenticator and

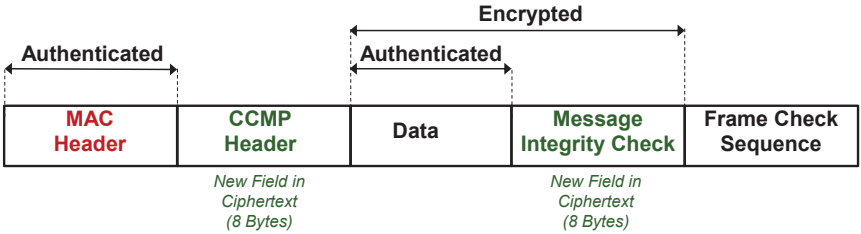

Figure 3.3: Overview of the output of CCMP.

once as supplicant. Prior to the handshake, both communicating parties derive a master key from a pre-shared password using a static non-interactive method. During the handshake, each party proves their knowledge of the master key and thus of the pre-shared password, see [Joh10], [Liu08], or [IEE04]. In the case of IEEE 802.11s, the SAE protocol is used for the mutual authentication of nodes. It is a password-based authentication protocol that considers the involved parties as peers, see [Egn10, Har08]. SAE does not assume a knowledge of a master key. In contrast, the output of the protocol is the master key. Here, the communicating parties prove their knowledge of a pre-shared password and interactively derive the master key using a zero knowledge protocol, i.e., without revealing any information about the key or the password. Consequently, SAE is more resilient to offline dictionary attacks than the handshake mechanism in IEEE 802.11i, see [IEE11].

3. *Key Derivation*: This phase is accomplished as part of the four-way handshakes that occur with IEEE 802.11i. In the case of IEEE 802.11s, the Authenticated Mesh Peering Exchange protocol (AMPE) is used. In both cases, a common pairwise key for unicast frames and two group keys (one for each communicating party) for broadcast frames are derived. In both cases, eight messages are exchanged in total.

   Security limitation in the mesh backbone. Both frameworks are based on static passwords, without supporting a dynamic refresh of the password. Only the derived keys might be refreshed on the fly. Hence, once the attacker compromises the password, which is the essential security credential, the attacker is able mount all kinds of internal routing attacks. It is shown in Section 7.2 how the attacker can exploit this issue to launch a blackhole attack.

4. *Protected Data transfer*: This is the most relevant phase with respect to external routing attacks. Both frameworks use the CCMP protocol for the

confidentiality and integrity of the frames. The output of CCMP (i.e., the cipher text) is illustrated in Figure 3.3.

Security limitation in the mesh backbone. The MAC header is authenticated but not encrypted. This means that an external attacker cannot change the header but can read it. Consequently, an external attacker can successfully replay a frame at another location by manipulating its own MAC address to match that of the frame. This issue can be misused to implement the wormhole attack, as shown in Section 7.2.

3.2 Deployment Impediments of Secure Routing Proposals

In the last decade, a profusion of approaches have been proposed to secure the routing process in WMNs [Sgo13, Sen13, Abu08, Hu04]. However, none of the existing proposals has gained acceptance in practice, and so, research in this area is still very active. In this section, the most well-known and recent proposals are reviewed (see Table 3.2), and the main reasons that have burdened their deployment are elaborated. Here, individual shortcomings and vulnerabilities related to specific scenarios and security goals are not covered in details (as in [Sen13, Abu08]), the focus rather lies on common feasibility-related limitations. As Table 3.2 shows, the existing secure routing approaches can be mainly divided into two classes: being based on symmetric-key or on asymmetric-key cryptosystems. Surprisingly, to the best of the knowledge of the author of this thesis, none of the existing approaches implements a hybrid security scheme, even though some approaches support both cryptosystems.

3.2.1 Asymmetric-Key-Based Secure Routing Proposals

In asymmetric-key-based secure routing proposals, such as ARAN, Secure Optimized Link State Routing (SOLSR) [Hon05], Secure Ad-hoc On-demand Distance Vector (SAODV) [Zap02], and Secure Wireless Mesh Protocol (SWMP) [Mat14], a Public key Infrastructure (PKI) is assumed, with each node having a key pair and a certificate. In UAV-WMN, this assumption is feasible as it can be realized by the network operator implementing the certification authority. In IBC-HWMP [BO13] and IBC-Radio Aware OLSR (RAOLSR) [SB14], Identity Based Cryptography (IBC) is proposed to avoid the need for a PKI. However many issues in IBC are still unsolved [Zha12], besides, IBC schemes are typically based on Elliptic Curves Cryptography (ECC), which is also used in Elliptic Curve Digital Signature Algorithm (ECDSA)-RAOLSR [SB14], and the information leaked in 2013 by Edward Snowden revealed that standardized ECC-based algorithms were influenced to include backdoors [Ber14b]. Apart from that, due to the

Table 3.2: Selected list of well-known as well as recent secure WMN routing proposals [Sbe15].

Protocol name	**Cryptosystem class**	**Main security techniques**	**Deployment impediment in UAV-WMN**
ARAN [San05]	Asymmetric-key	Digital signature (PKI)	Computationally expensive on embedded systems [Kno13, Wol03]
IBC-HWMP [BO13]		Digital signature (IBC), neighbor monitoring	
IBC/ECDSA-RAOLSR [SB14]		Digital signature (IBC/PKI-ECC)	
SAODV [Zap02]		Digital signature (PKI), hash chain	
SOLSR [Hon05]		Digital signature (PKI), hash chain, temporal leash	
SWMP [Mat14]		Digital signature (PKI)	
SEAD [Hu03a]	Symmetric-key	*MAC*, hash chain, Merkle tree	Interdependency cycle with dynamic key distribution methods [Zha13b, Bob03]
SHWMP [Isl09]		*MAC*, Merkle tree	
SEAODV [Li11]		*MAC*	
Ariadne [Hu05]	Symmetric-key	*MAC* (or digital signature), hash chain	
Castor [Gal10]	(or asymmetric)	*MAC* (or d. s.), Merkle tree, PDR per flow	

ECC: Elliptic Curve Cryptography, IBC: Identity Based Cryptography, PDR: Packet Delivery Ratio, PKI: Public Key Infrastructure

complexity of ECC, well known cryptographers have implementation concerns, which could make the system vulnerable despite the security of the algorithm [Tan15, Eme15]. Moreover, some cryptographic operations are still faster using RSA than using ECC, cf. signature verification in [Wol03].
To provide origin authentication, message freshness, and message authentication, asymmetric-key-based secure routing proposals use nonces (random values or counters) and digital signatures. For instance, let O be the originator, I_1 the first intermediate node, I_2 the second one, and D the destination. Let $RREQ$ denotes relevant fields of a route request, and $addr$ be an IP or a MAC address. If O wants to discover a route to the destination, it broadcasts the following:

$$RREQ_O := (addr_O, addr_D, nonce_O)_{sig_{k_{priv_O}}}$$

When I_1 receives the message, it makes sure, based on the certificate, that the key used to sign the message is that of O (i.e., origin authentication), it checks if the nonce is fresh (i.e., message freshness), and it verifies the signature of the message (i.e., message authentication). Afterwards, I_1 signs the message, appends its certificate to it, and rebroadcasts it.

$$RREQ_O := (((addr_O, addr_D, nonce_O)_{sig_{k_{priv_O}}})_{sig_{k_{priv_{I_1}}}}, cert_O, cert_{I_1})$$

When I_2 receives the message, it verifies the freshness of the message, and the identity and signature of both the originator and the sender (i.e., the previous hop I_1). It then replaces the signature and certificate of the previous hop I_1 with its own, and it resends the message.

$$RREQ_O := (((addr_O, addr_D, nonce_O)_{sig_{k_{priv_O}}})_{sig_{k_{priv_{I_2}}}}, cert_O, cert_{I_2})$$

Each following intermediate node along the route repeats the same steps as I_2.
To guarantee neighbor authentication, SOLSR also uses packet leashes, which are proposed in [Hu03b] to combat the wormhole attack. Packet leashes are either geographical or temporal.

- Geographical leashes: Position information is included in the routing messages so that the receiver can make sure, based on the distance, that the sender is within its transmission range. This approach only works well in environments where there are no physical obstacles preventing the communication between nodes, e.g., in UAV-WMN.
- Temporal leashes: Timestamps are included in the routing messages, and if the transmission time of a message exceeds a threshold, the message is dropped (e.g., SOLSR is based on such a scheme). When implementing this approach, all nodes must be accurately synchronized with respect to time in terms of microseconds, which is not straightforward in practice. Besides, the scheme does not take into account the channel access delay at the

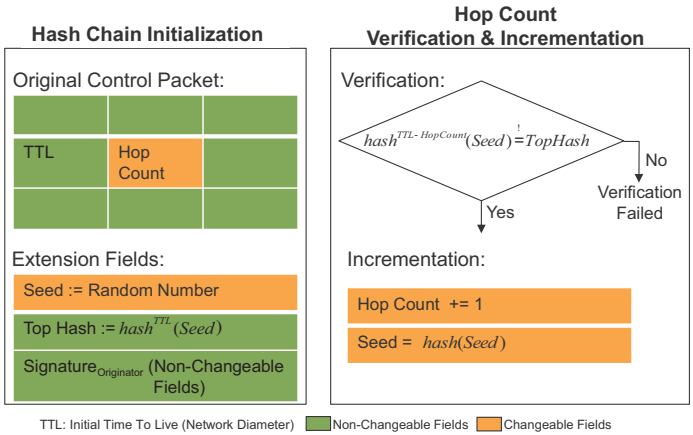

Figure 3.4: Overview of the hop authenticator method for securing the hop count.

MAC/PHY layers, resulting from Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA).

To react to the compromise of a key or node, ARAN supports dynamic key revocation, i.e., revoking the corresponding certificate and flooding it in the network. However, the effectiveness of this approach is limited, as it does not take into account unsuccessful receptions of the revocation message due to collisions or because some nodes are instantly disconnected.

To minimize the harm that can be done by an internal attacker, a couple of approaches (e.g., SAODV, SOLSR) include a hop authenticator in the routing messages, as illustrated in Figure 3.4. They use a hash chain to prevent a malicious intermediate node from decrementing the hop count in order to persuade other nodes that the route that goes through the malicious node is the shortest one. A hash chain is a sequence of hash values computed by iteratively calling a cryptographic hash function H to an initial value v_0, i.e., $v_{i+1} = H(v_i) = H^{i+1}(v_0)$. For example, $v_4 = H^4(v_0) = H(H(H(H(v_0))))$. Since a cryptographic hash function is a one way function, a malicious intermediate node that is n hops away from the message originator will only be aware of the n^{th} element in the hash chain, but not of any element that comes before. Thus, the attacker cannot decrease the hop count, otherwise the verification in Figure 3.4 fails. However, this scheme is only effective to a small extent, because it can only be used in coordination with the hop count, i.e., it is, unfortunately, useless in case other metrics such as the airtime link metric [IEE11] are implemented. Besides, the attacker can still forward the message without increasing the hop count.

To detect internal malicious nodes, IBC-HWMP proposes to monitor the behavior of the neighbors. This requires an extra interface in monitor mode, which is very critical in UAV-WMN due to the limited size and weight of the UAVs. Additional limitations of neighbor monitoring are provided in [Abu08].

Deployment impediment of asymmetric-key-based secure routing proposals. In this class of proposals, every node (i.e., originator or intermediate) must sign every message. This provides a high level of security, but it has a high computation time in UAV-WMN, where embedded systems are used [Kno13, Wol03]. For instance, digital signature operations using RSA-1024 and EDCSA-160 take longer than 26 ms on the Roboard RB110 [ROB] (x86, 1 GHz, 256 MByte DRAM). This holds for 30 measurements executed using ftrace [Bir09]. Thus, in case of a route with five intermediate hops, the delay is higher than 156 ms. This does not satisfy the quality of user experience of multimedia streaming, where according to [Abo03] the delay should be below 150 ms —relying on graphical processing units to address this issue does not solve the problem as the parallelism of one digital signature operation comes with the disadvantages of thread synchronization and data exchange overhead [Sch14].

3.2.2 Symmetric-Key-Based Secure Routing Proposals

In contrast to the high processing time of asymmetric-key-based secure routing messages (especially due to digital signatures), that of secure routing messages based on symmetric cryptosystems is relatively low, e.g., as in the secure on-demand routing protocol Ariadne [Hu05], Continuously Adapting Secure Topology-Oblivious Routing (CASTOR) [Gal10], Secure Efficient Ad-hoc Distance vector (SEAD) [Hu03a], Security Enhanced AODV (SEAODV) [Li11], and Secure HWMP (SHWMP) [Isl09]. These proposals mainly use cryptographic hash-function based techniques to secure the routing messages. The cost of SHA-256 is below 0.15 ms, based on 30 measurements using ftrace and 1500 random bytes. The cost of running 20 iterative calls of the hash function is below 0.20 ms. That is, iteratively calling the hash function only causes a relatively slight increase in the time cost as the instructions and data are already in the processor cache and because the hash function operates efficiently. The cryptographic hash function-based techniques used by this class of routing proposals include *MAC*, hash chains, and Merkle trees. To avoid ambiguity, *MAC* (in italics) always refers to a message authentication code while MAC denotes medium access control.
Based on the assumption that the nodes share pairwise secret keys, all the proposals use *MAC* for message authentication, either in an end-to-end fashion, such as in Ariadne [Hu05] and Castor [Gal10], or in a hop-by-hop fashion, such as in SEAD [Hu03a], SWHMP [Isl09], and SEAODV [Li11]. *MAC* can be seen as the

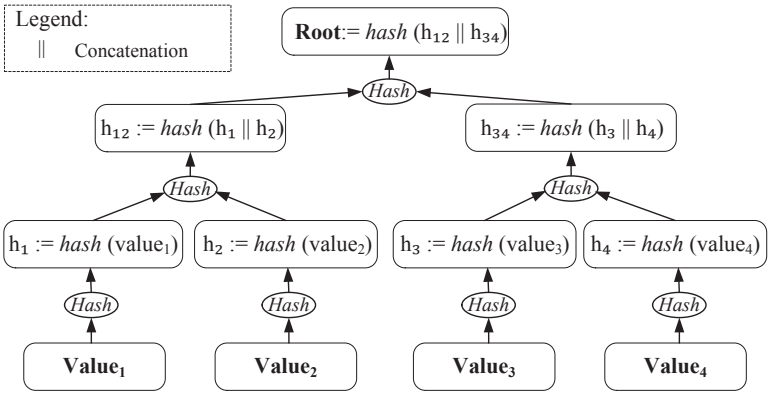

Figure 3.5: Example of a Merkle tree with four leafs.

pendant of digital signature for symmetric-key based secure routing proposals. It is typically used with nonces to ensure message freshness.
To minimize the harm that can be done by an internal attacker, e.g., to prevent manipulating the list of forwarding nodes, hash chains are used. For instance, let I_1, I_2, and I_3 be successive intermediate nodes between O and D. When these nodes forward $RREQ_O$, they append their identity to the message, and they build a hash chain on it to prevent an attacker from manipulating or removing this information. As an example, I_3 will forward $RREQ_O$ as follows:

$$RREQ_O := (addr_O, addr_D, nonce_O)_{MAC_{k_{OD}}}, (I_1, I_2, I_3), v_3)$$
$$v_3 = H(I_3, H(I_2, H(I_1, MAC_{k_{OD}}(addr_O, addr_D, nonce_O))))$$

It has been shown that the security of this mechanism is limited, as it is vulnerable to manipulation, especially in dense networks where the attacker receives $RREQ_O$ via different routes and thus can know different hash elements in the hash chain [Bur07].
To further minimize the harm that can be done by an internal attacker, Merkle trees are used. A Merkle tree is a perfect binary tree composed of 2^n leafs with $n \in \mathbb{N}$, where each internal node in the tree is a hash of its two child nodes. Figure 3.5 illustrates an example of a Merkle tree consisting of four leafs. Merkle tree is employed differently in different proposals. In SEAD, Merkle trees are integrated into the hash chains to prevent the attacker from forwarding the messages without adding its identity or incrementing the hop count (e.g., when using a hop authenticator). Briefly, the idea can be described as follows: let v_i and v_{i+1} be two successive elements in the hash chain. Usually, a node that receives

v_i in a routing message computes and forwards v_{i+1} by hashing v_i with some information. In SEAD, each element in the hash chain (e.g., v_i or v_{i+1}) is a root of a Merkle tree, see Figure 3.5 for an example of a Merkle tree. Here, the leafs of the tree are computed based on the previous element in the hash chain and the identity of the nodes in the network. For instance, the leafs of the Merkle tree having v_{i+1} as root are $l_1 = H(v_i||ID_{node_1})$,..., $l_n = H(v_i||ID_{node_n})$ with n being the number of nodes in the network. When using this scheme, the nodes do not forward the hash element, i.e., v_{i+1}, but they forward the leaf corresponding to their identity and some additional information in the tree needed to compute the root. Now assume that $node_1$ forwards l_1 and the corresponding tree information to compute v_{i+1}. An attacker that receives this message cannot just pass the message to the next-hop because l_1 does not match its identity, and it cannot generate a leaf other than l_1 as it does not have knowledge of v_i. Thus, the attacker has to include new information to forward the message. As a matter of fact, this scheme is only effective (from the security point of view) if the attacker cannot fake the identity of other nodes.

Merkle trees are used in CASTOR in another context. The protocol not only deals with secure route discovery, but also with secure data forwarding (i.e., with the evaluation of the routes), and it uses the packet delivery ratio at the network layer as a security metric. To this end, for each packet sent from source to destination, the destination sends back a secure ACK via the reverse route. The forwarding nodes monitor the PDR for each flow and decide, based on this metric, whether the routes are reliable or not. Thus, nodes need to securely identify which packets belong to which flow to derive reliable decisions. Here, Merkle trees are used. The source of each flow pre-generates a large number of random values. These random values build the leafs of the Merkle tree, and they are kept secret. The root of the tree becomes the flow identifier, and it is announced during the route discovery. Whenever the source sends a data packet belonging to the flow, it includes the flow identifier, one of the random values, and some tree information to validate that the random value has been used to compute the flow identifier. In this way, the intermediate nodes can always securely match the packets to a flow identifier, as only the source has knowledge of the random values. While this scheme is effective from the security point of view, it is questionable whether it is efficient from the performance point of view in terms of tree generation and storage, i.e., the number of trees and number of random values.

In SHWMP, Merkle trees are used in combination with MAC and the key scheme of IEEE 802.11s to authenticate the mutable fields in a routing message, in a hop-by-hop fashion. In the authors' opinion, this combination does not improve the security of the protocol as using MAC and the key scheme of IEEE 802.11s already leads to one-hop message authentication.

Deployment impediment of symmetric-key-based secure routing proposals. This class of proposals requires that for every route discovery, the source and destination (and neighbors) must have a security association between them. That is, the existence of a real-time key distribution method is assumed. This is not straightforward in WMN [Cha05b]. In turn, to dynamically distribute or revoke symmetric keys, reliable routes between the nodes are required [Zha13b, Leb13]. In other words, there is an interdependency cycle problem [Bob03].

In comparison to the first secure routing method, which relies on the IEEE 802.11s/i security frameworks, the proposed solution in this thesis (PASER) provides better routing security in UAV-WMN. In comparison to existing secure routing proposals in the literature, PASER implements a hybrid security scheme that takes the specifics of the target network (i.e., UAV-WMN) into consideration, and it incorporates a key distribution method. Hence, it is more efficient than asymmetric-key-based proposals, and it is not based on unrealistic assumptions, as opposed to the symmetric-key-based proposals.

4

On the Credibility of Simulating Highly Dynamic WLAN Mesh Networks

Contents of this Chapter

This chapter discusses the credibility of evaluating the performance of WLAN mesh routing protocols in the discrete event-based network simulator OMNeT++ and its INETMANET framework. In this regard, the simulation methodology applied in this research to draw valid conclusions is elaborated.
A brief introduction to OMNeT++ is given in Appendix B. This tool is used in this research for the following reasons:

- It is released under an open source license for academic use, allowing for extending existing models and adding new protocols at no charge.
- It implements a hierarchical architecture to build the simulation model (e.g., interface–nodes–networks). It has a modular concept, i.e., simple modules are nested inside each other to form a complex module that defines a component of the simulation model, e.g., a node. These characteristics enable a straightforward development and integration of new protocols without requiring a precise knowledge of the whole system.
- It uses an easy to learn text-based language to define the modules that build the simulation model, and the behavior of these modules is implemented in the well-known C++ language. This enables using OMNeT++ without

needing to learn complex languages, and this increases the portability of the simulation code.

- It features a friendly yet advanced graphical user interface. This interface allows debugging simulations during runtime with very high granularity, which speeds up the development and analysis of new research approaches.
- It runs on Unix-based systems and on Windows. It is well-documented, constantly maintained, widely supported, and it includes the INETMANET framework, which implements all standards and routing protocols that are relevant to accomplish this thesis.

A profusion of simulation models that use OMNeT++ have been developed so far. This has led to the formation of various frameworks that cater to domain-specific models, see [Var10]. Examples of well-known OMNeT++ simulation frameworks are:

- INET Framework [INEa]: It is one of the largest OMNeT++ network simulation frameworks. It contains models for various wired and wireless networking protocols such as UDP, TCP, SCTP, IP, IPv6, Ethernet, OSPF, IEEE 802.11, and others.
- INETMANET Framework [INEb]: It is a fork of the INET Framework with special focus on the simulation of wireless multi-hop networks. Example models of INETMANET include propagation models, wireless link layer protocols, and WLAN mesh routing protocols such as OLSR, BATMAN, AODV, and HWMP.

In this research, INETMANET is used as the main platform to investigate the performance of (secure) routing protocols in highly dynamic WMNs. An overview of modeling WMNs in INETMANET is given in Appendix C.

4.1 Credibility of Simulating WMNs in OMNeT++

Simulation provides an attractive method of evaluating communication networks, due its high degree of controllability and repeatability. Simulation studies are flexible to construct, and the associated costs are typically low. This is especially advantageous when studying UAV-WMN, as deploying such a network entails high operational costs, and because UAVs in practice are currently subject to restrictions by regulations and civil security concerns. Besides, the results of test beds are not always reproducible.
In comparison to mathematical models that do not cover UAV-WMN as a whole but just parts of it, due to the complexity of combining the different aspects of UAV-WMN, e.g., radio propagation, interference, protocol interactions and

others, various comprehensive frameworks of WMNs such as INEMANET exist in simulation.
Nevertheless, simulation studies are fraught with pitfalls. Without a clear definition of the goals of the simulation, a deep knowledge of the simulation model used, and a validation of the abstractions implemented in the model with respect to the defined goals, simulation studies might lead to the wrong conclusions. They could produce misleading results that do not reflect the reality. In the following, common simulation errors are listed, and the procedure to generate valid and credible results in this research is elaborated.

Common Flaws of Simulation Studies. According to the surveys in [And06, Kur05] on the credibility of simulation studies on wireless multi-hop networks, common shortfalls are the following:

- *Non-repeatable simulations* due to the absence of significant information, such as the code of the model, important simulation settings (e.g., simulation time, relevant protocol parametrization), the version of the simulation tool and framework in use, and others.
- *Lack of model validation* because simulated models (or enhancements of existing models) are rarely validated against either the corresponding experimental implementations or through analytical methods.
- *Unrealistic scenario design* in terms of the channel model (e.g., just using free-space), traffic types, node mobility, transmission ranges, size of the simulation area, and others.
- *lack of rigor* by generalizing results produced by one single scenario or by using the default parameters of the simulation model, where many of them have been configured to meet generic requirements.
- *Improper output analysis* due to the absence of independent simulation runs and confidence intervals. Besides, simulation assumptions and results are hardly justified.

4.2 Applied Simulation Methodology

To overcome the simulation shortfalls presented in Section 4.1, the following steps are applied in this thesis to study highly dynamic WMNs in simulation using INETMANET.

1. *Identification of the limitations (i.e., deviations from practice) of the WMN model in INETMANET.* The most relevant deviations are:

Table 4.1: Important features of the PHY layer of relevant IEEE 802.11 releases.

Name	Title	Release date	Frequency [MHz]	Bandwidth [MHz]	PHY data rates per stream	Mandatory PHY data rates	MIMO streams
802.11-1997	IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications	1997	2.4	20	1, 2	1, 2	1
802.11a	Higher Speed PHY Extension in the 5 GHz Band	1999	5	20	1, 6, 12, 18, 24, 36, 48, 54	6, 12, 24	1
802.11b	Higher Speed PHY Extension in the 2.4 GHz Band	1999	2.4	20	1, 2, 5.5, 11	1, 2, 5.5, 11	1
802.11g	Further High Data Rate Extension in the 2.4 GHz Band	2003	2.4	20	those of 802.11a & 802.11b	those of 802.11a & 802.11b	1
802.11n	Enhancements for Higher Throughput	2009	2.4, 5	20, 40	up to 72.2 @ 20 MHz, 150 @ 40 MHz	those of 802.11a & 802.11b	4
802.11ac	Enhancements for Very High Throughput for Operation in Bands Below 6 GHz	2013	2.4, 5	20, 40, 80, 160	up to 86.7 @ 20 MHz, 200 @ 40 MHz, 433.3 @ 80 MHz, 866.7 @ 160 MHz	those of 802.11a & 802.11b	8

Table 4.2: IEEE 802.11 frame types.

Type	Relevant example frames
Data	Data
Control	Request-To-Send (RTS) Clear-To-Send (CTS) Acknowledgment (ACK)
Management	Beacon

- INETMANET does not include a model of recent WLAN standards such as IEEE 802.11n [IEE11] or IEEE 802.11ac [IEE13a], which offer PHY data rates up to 800 Mbit/s in the case of 2x2 MIMO and channel bundling (40 MHz), see Table 4.1. It rather implements the IEEE 802.11a/b/g standards, which have maximum PHY data rates of 54 Mbit/s, as depicted in Table 4.1. Thus, INETMANET should not be used to evaluate the capacity-related issues of WMNs. For instance, it should not be used to investigate the maximum sustainable network throughput (i.e., saturation throughput) for a given scenario in practice. Besides, special care should be given to the parametrization of the modeled network (e.g., application data rates) in order to not always have an overload situation. Here, according to the IEEE 802.11g standard [IEE03], the nodes responding to a received frame shall transmit their control response frame (e.g., ACK) at the highest rate in the set of the mandatory PHY data rates (i.e., 1, 2, 5.5, 6, 11, 12, and 24 Mbit/s) that is less than or equal to the rate of the received frame and that is of the same modulation class as the latter, see Table 4.2 for an overview of the different frame types. While this statement holds in practice, in INETMANET, control frames (e.g., ACK) are always transmitted with a static PHY data rate of, e.g., 1 Mbit/s. This leads to different saturation throughputs in simulation from those in practice. Validating the simulation results in practice only makes sense when properly calculating and considering the different throughputs; more details about this point are provided in Sub-section 4.3.1.

- INETMANET does not implement the WLAN rate control algorithms that are currently used in practice, such as Minstrel [MIN]. It implements the Auto Rate Fallback (ARF) and the Adaptive Auto Rate Fallback (AARF) mechanisms [Lac04]. ARF was the first rate adaptation published for WLAN, in 1997. AARF is an optimization of ARF, proposed in 2004 to reduce unnecessary back-and-forth rate switches. Both mechanisms are early solutions and are no longer supported in practice, see [Xia13, Xia12]. Besides, while PHY data rates are sender-

receiver specific in practice, in INETMANET they are only related to the sender. That is, a sender uses only one PHY data rate to send its frames, regardless of the receiver of the frame and the channel characteristics of the latter. Thus, for the analysis of WMNs' routing protocols in INETMANET, it is more reasonable to set a static PHY data rate to avoid generating ambiguous results influenced by a rate control algorithm that is not valid and has no relevance in practice. The reader is referred to [Bic05b] for information about the impact of rate adaptation mechanisms on the performance of wireless mesh networks.

- *Verification of the WLAN model in INETMANET.*
  Several papers exist on the validation of different aspects of the WLAN model in INETMANET. These include the validation of throughput (IEEE 802.11 DCF behavior in case of a variable number of contending clients) [Bre09], the impact of the Signal-to-Noise Ratio (SNR) on the throughput and frame error rate in the case of a static PHY data rate of 12 Mbit/s [Chl12], handover delay (a WLAN client moving between two access points) [Woo03], and radio propagation models [Kun08]. While all these papers attest to the validity of the analyzed aspects of the WLAN model in INETMANET, especially in the case of long observation times and on-average behavior, the authors of [Bre09] observe that the distribution of inter-transmissions in the case of three and four contending stations differs from that of the measurements in a testbed. They conclude that the packet scheduling differs between the testbed and the simulation. To cope with this discrepancy and to verify whether changes made in the new INETMANET version (2.0) are still valid, the IEEE 802.11 code in the framework were crosschecked, and its conformity with the IEEE 802.11 standard was verified. Different simple scenarios were analyzed (e.g., throughput in the case of two or three nodes having different distances, PHY data rates, and application data rates) and the results were compared with those of an own MATLAB [MAT11] implementation of the corresponding functions of the IEEE 802.11 standard, see Sub-section 4.3.1. The author of this thesis was fortunate to find the bug that caused the aforementioned discrepancy as well as another crucial error, see [BUG13a] and [BUG13b], respectively.

- *Close-to-reality parametrization of the INETMANET-WMN model and testbed-based validation of relevant routing protocols models.*
  A WMN testbed based on IEEE 802.11g was set up for validation purposes, see Sub-section 4.3.2. The parameters of the WLAN model in INETMANET were changed to match those of the IEEE 802.11g standard (e.g., frequency, slot duration, contention window size, etc.).

The WLAN interface settings were adjusted to reflect the capabilities of the WLAN device used in the WMN testbed. The code of relevant routing protocol models (i.e., message size, processing time of important functions, behavior) was reviewed to match that of real protocols in the testbed, see Chapter 8. Besides, the routing protocols were configured to have a similar performance to that of the real protocols in the testbed, see Sub-section 4.3.3.

- *Formulation of simulation goals that consider the limitations of INETMANET while addressing realistic scenarios to generate valid results.*
  The main purpose of simulation in this thesis is to compare the performance of a proposed secure routing protocol for highly dynamic WMNs with that of well-established routing protocols. Here, the focus lies on the comparative analysis of the impact of UAV mobility on the analyzed solutions as well as the relative scalability of these solutions in terms of node density and route length. To achieve this goal, the PHY layer of IEEE 802.11g is used as a reference technology, and realistic mobility patterns of UAVs [God12a, God14, Beh13] and a valid air-to-air radio propagation model are considered, see Chapter 8.

- *Running multiple runs using a Mersenne Twister Pseudo Random Number Generator (PRNG) and different seeds to generate independent and identically distributed results.*
  For evaluating the performance of the routing protocols, at least five simulation runs and up to twenty runs were executed to generate the results, and a confidence interval of 97.5 % using Student's t-distribution (unless differently stated) is used to interpret the results. For analyzing the behavior of the protocol, however, only selected results of the packet delivery ratio versus time are considered, as the purpose in this case is to depict the impact of a given aspect on the behavior of the protocols.

- *Published extensions added to INETMANET including the code of the proposed secure routing protocol.*
  To ensure the repeatability of the simulation studies in this thesis, the code of the models used (i.e., the new models and the extensions of existing models), the corresponding parameterizations, and further relevant information are either integrated in the official INETMANET framework or/and are published on `wwww.paser.info`.

4.3 Validation of WLAN Mesh Routing Models in OMNeT++

In this section, the models of the routing protocols HWMPS and BATMANS in OMNeT++-INETMANET and the underlying WMN model are validated with respect to performance as these are used to assess the efficiency of the secure routing approach proposed in this thesis. In this regard, the link throughput and the network saturation throughput are calculated in theory in case of two hops (no collision), and it is verified whether the simulation results equal the theoretical ones. A perfect matching is obtained, as shown in [Sbe14c]. As a further step, the network saturation throughput is estimated in case of five hops (collision), and simulation and experimental measurements are performed using the estimated network saturation throughput as the traffic load. A comparison of the results, discussed in this section, show a high similarity between simulation and experiments.

4.3.1 Theoretical Estimation of Network Saturation Throughput

In this sub-section, the link capacity and the network saturation throughput of WMNs are analytically estimated —for an accurate analysis of the performance results and for a fair simulation configurations.
The network saturation throughput is defined as the maximum sustainable throughput [Bia98]. It is the maximum load that the network can carry in stable conditions. According to [Ng07], operating WLAN muli-hop networks under this throughput prevents instability problems, such as throughput oscillations and link-failure triggered re-routing. The authors of [Ng07] prove this fact based on a quantitative analysis. They also provide a closed-form analytical solution to calculate this throughput, assuming the knowledge of hard-to-compute, non-measurable parameters, such as the collision probability for a transmission. Bearing in mind that the saturation throughput of a WMN corresponds to the throughput of the bottleneck link in the network topology [Zha13a], a simplistic method is used in this sub-section to estimate this throughput by only relying on easy-to-get measurable information, such as the network topology and the links PHY data rate. Here, the following two steps are carried out:

1. The throughput of each WMN link is calculated under the assumption that no other links exist in the network (no collision);

2. The bottleneck link in the network is determined based on the link throughput information and a contention graph of the network, cf. [Che05].

Table 4.3: Notations used in IEEE 802.11 throughput equations.

Notation	Description	Notation	Description
ACK	Acknowledgment frame	Min	Minimum
$Bcst$	Broadcast	Ohd	Overhead
CTS	Clear-To-Send	PHY	Physical layer
CW	Contention Window	RTS	Request-To-Send
DR	Date Rate	$SIFS$	Short InterFrame Space
$DIFS$	Distributed coordination function Interframe Space	T	Time
Hdr	Header	TCP	Transmission Control Protocol
IP	Internet Protocol	Thr	Throughput
MAC	Media Access Control	$Ucst$	Unicast
Mdt	Mandatory rate	UDP	User Datagram Protocol

4.3.1.1 Calculation of the Link Throughput

The link throughput corresponds to the link capacity. It is the number of sent bits per second at the PHY layer. The throughput of an IEEE 802.11 link can be calculated according to Equation 4.1 (see Table 4.3 for the definition of the notations).

$$Thr\ [\text{bit/s}] = \frac{Size_{Frame}}{T_{TX_{Frame}}} \tag{4.1}$$

According to the standard [IEE13b], this throughput depends on the following factors:

- Frame size: The size of a frame is calculated according to Equation 4.2.

$$\begin{aligned} Size_{Frame}\ [\text{bit}] = Size_{Payload} + Size_{Hdr_{UDP/TCP}} \\ + Size_{Hdr_{IP}} + Size_{Ohd_{MAC}} \end{aligned} \tag{4.2}$$

- Transmission time of the frame: This time varies mainly based on:
  - The carrier sense mechanism

– The destination type of data frames (unicast or broadcast)

– The PHY data rate.

In IEEE 802.11, two mechanisms are defined for carrier sensing of the shared wireless medium.

1. The first mechanism uses physical carrier sensing to ensure that the medium is free. Using this mechanism, only nodes that are in the carrier sense range of the sender detect its transmission, but those that are outside this range, yet inside the range of the receiver, will not be aware that the channel is busy. This could lead to collisions at the receiver in the case of simultaneous transmissions.

2. The second mechanism additionally uses virtual carrier sensing to indicate an idle channel. The IEEE 802.11 standard specifies the Request-To-Send (RTS)/Clear-To-Send (CTS) mechanism for virtual carrier sensing [Phi90]. The goal of this mechanism is to mitigate the hidden station problem, which is defined as the interference caused by simultaneous transmissions of hidden nodes because one receiver is in the range of at least two senders [Ful97]. To achieve this goal, an RTS/CTS frames handshake is completed between the sender and the receiver before starting the data exchange. This handshake informs nodes that are in the proximity of both the sender and the receiver that the channel is busy. It also gives them information about the time to hold off from accessing the medium until the transmission is finished. The RTS/CTS mechanism is only applied to unicast data frames. In the case of broadcast frames, only the basic mechanism is used, as there is no specific frame destination necessary for the exchange of RTS/CTS.

According to the EDCA mechanism [IEE13b], the average transmission time of a unicast data frame using either physical carrier sensing (termed hereafter Basic) or virtual carrier sensing (termed hereafter CTS) are provided in Equation 4.3 and Equation 4.4, respectively.

$$T_{TX_{Ucst_{Basic}}}\ [\mathrm{s}] = AIFS + \frac{(CW_{min} - 1) \cdot T_{Slot}}{2} + T_{Data_{Ucst}} + SIFS + T_{ACK} \tag{4.3}$$

$$\begin{aligned} T_{TX_{Ucst_{CTS}}}\ [\mathrm{s}] = AIFS + \frac{(CW_{min} - 1) \cdot T_{Slot}}{2} + T_{RTS} + SIFS + \\ T_{CTS} + SIFS + T_{Data_{Ucst}} + SIFS + T_{ACK} \end{aligned} \tag{4.4}$$

Broadcast frames have transmission times that are different from those of unicast frames. This time is calculated using Equation 4.5: According to the standard, unicast frames are acknowledged, and they shall be transmitted at the highest supported PHY data rate that is suitable to the link characteristics (e.g.,

Table 4.4: Relevant IEEE 802.11g configurations.

$\boldsymbol{T_{Slot}}$ ERP-OFDM: 9 μs DSSS: 20 μs	$\boldsymbol{SIFS}$ 10 μs	$\boldsymbol{AIFS}$ $3 \cdot T_{Slot} + SIFS$ (best effort*)	$\boldsymbol{CWmin}$ 31
$\boldsymbol{Ohd_{MAC}}$ 320 bit	$\boldsymbol{Hdr_{PHY}}$ ERP-OFDM: 20 bit DSSS: 192 bit	$\boldsymbol{ACK}$ 112 bit	$\boldsymbol{RTS}$ 160 bit
$\boldsymbol{CTS}$ 112 bit	$\boldsymbol{DR_{PHY_{Data}}}$ ERP-OFDM: 6, 9, 12, 18, 24, 36, 48, 54 Mbit/s DSSS: 1, 2, 5.5, 11 Mbit/s	$\boldsymbol{DR_{PHY_{Mdt}}}$ $f(DR_{PHY})$	$\boldsymbol{DR_{PHY_{Bcst}}}$ 1 Mbit/s (default value)

*: EDCA best effort is the default priority class in practice.

48 Mbit/s, 54 Mbit/s in IEEE 802.11g). In contrast, broadcast frames are not acknowledged, and they are transmitted with one of the mandatory rates that are supported by all WLAN devices (e.g. , min. 1 Mbit/s, max. 24 Mbit/s in IEEE 802.11g). In case of IEEE 802.11g in practice and in most simulation configurations, these frames are typically transmitted with 1 Mbit/s.

$$T_{TX_{Bcst}}\ [\text{s}] = AIFS + \frac{(CW_{min} - 1) \cdot T_{Slot}}{2} + T_{Data_{Bcst}} \tag{4.5}$$

The impact of the PHY data rate on the transmission time of unicast, broadcast, and control frames, is reflected in Equation 4.6, Equation 4.7, and Equation 4.8, respectively. In the case of unicast data frames, the highest suitable data rate is used. In the case of broadcast frames, a fixed low data rate (robust against channel errors) is used. In the case of control frames (ACK, RTS and CTS), the mandatory PHY data rates are used.

$$T_{Data_{Ucst}}\ [\text{s}] = \frac{Hdr_{PHY}}{1 \cdot 10^6\ \text{bit/s}} + \frac{Size_{Frame}}{DR_{PHY}} \tag{4.6}$$

$$T_{Data_{Bcst}}\ [\text{s}] = \frac{Hdr_{PHY}}{1 \cdot 10^6\ \text{bit/s}} + \frac{Size_{Frame}}{DR_{PHY_{Bcst}}} \tag{4.7}$$

$$T_{ACK/RTS/CTS}\ [\text{s}] = \frac{Hdr_{PHY}}{1 \cdot 10^6\ \text{bit/s}} + \frac{Size_{ACK/RTS/CTS}}{DR_{PHY_{Mdt}}} \tag{4.8}$$

$$f(DR_{PHY})\ [\text{Mbit/s}] = \begin{cases} DR_{PHY}, & \text{IF} DR_{PHY} \in \{1, 2, 5.5, 11\} \\ 6, & \text{IF} DR_{PHY} \in \{6, 9\} \\ 12, & \text{IF} DR_{PHY} \in \{12, 18\} \\ 24, & \text{IF} DR_{PHY} \geq 24 \end{cases} \tag{4.9}$$

Table 4.5: Calculated IEEE 802.11g throughputs in case of 1460 Byte payload.

PHY data rate (data) [Mbit/s]											
1	2	5.5	11	6	9	12	18	24	36	48	54
PHY data rate (control) [Mbit/s]											
1	2	5.5	11	6		12		24			
PHY data rate (broadcast) [Mbit/s]											
1											
Link Throughput [Mbit/s], Basic, Unicast											
0.89	1.69	3.92	6.29	5.17	7.38	9.45	13.0	16.1	21.0	24.8	26.4
Link Throughput [Mbit/s], CTS, Unicast											
0.85	1.57	3.40	5.10	4.93	6.91	8.85	11.9	14.6	18.6	21.5	22.7
Link Throughput [Mbit/s], Broadcast											
0.91											

Using the parameter values illustrated in Table 4.4, the throughput of an IEEE 802.11g link is calculated in the case of different carrier sense mechanisms and frame destination types, and a 1460 Byte packet size at the application layer ($Hdr_{UDP} = 8$ Byte and $Hdr_{IP} = 20$ Byte). The results are depicted in Table 4.5. The table sheds light on two points. First, IEEE 802.11 has a poor support for broadcast applications. Second, the use of RTS/CTS for carrier sensing causes a non-negligible drop in throughput in comparison to the basic mechanism, especially in the case of high PHY data rates. This raises the question on whether it pays off to use this mechanism in wireless multi-hop networks (refer to Subsection 4.3.3 for more information).

*It should be noted that in OMNeT++-INETMANET the mandatory PHY data rate used to send control frames is not defined by Equation 4.9, it is rather fixed (*1 Mbit/s*, per default). Hence, discrepancy between simulation and practice might arise at high PHY data rates, even though, this discrepancy is compensated in multi-hop networks as the bottleneck link only achieves a small portion of the link throughput. Nevertheless, to compare the simulation results with those of a testbed, either the same conditions should be forced, or the differences in the PHY data rate should be taken into consideration.*

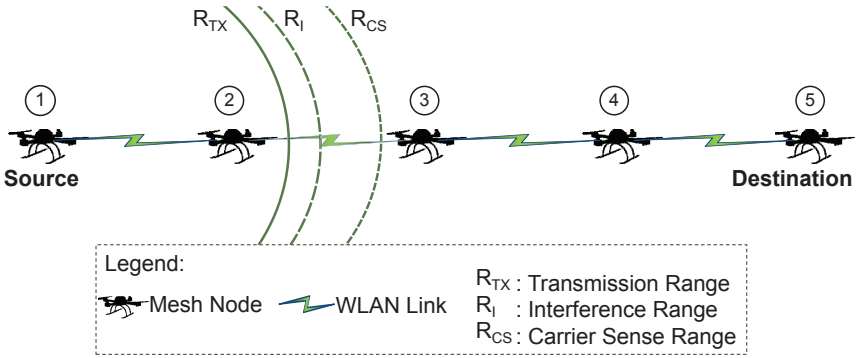

Figure 4.1: Setup of a static chain topology consisting of five nodes.

4.3.1.2 Determining the Bottleneck Link

To determine the bottleneck link in the network, the transmission probability P_{TX} of all network links is determined by using the contention graph method proposed in [Che05]. The contention graph accounts for medium access contention and for hidden nodes-based interference (i.e., interference between links). The resulting information aggregated with the link throughput information Thr reveals the bottleneck link.

Definition 1. *Let L be the set of links in the network: the bottleneck link is the link with the lowest value of $P_{TX_i} \cdot Thr_i,\ \forall i \in L$.*

For example, in case of a single flow consisting of two hops, $i \in \{1, 2\}$, it is $P_{TX_i} = \frac{1}{2}$, and the saturation throughput is $\frac{1}{2} \cdot min(Thr_i)$. The analysis in [Sbe14c] shows that the WMN model in OMNeT++-INETMANET produces a matching result in this straightforward scenario, in which no collision occurs as there is no hidden nodes. The estimation of the network saturation throughput becomes more complicated in case of hidden nodes, such as in the chain network of five nodes, which is depicted in Figure 4.1. The wireless links in that figure depict the transmission links of the nodes (all links are active as there is only one route to transmit the traffic flow between source and destination). The links also reflect the carrier sense range of the nodes (i.e., only Node 2 is in the carrier sense range of Node 1; this range ends between Node 2 and Node 3). To calculate P_{TX_i} in this network, it is necessary to translate the network topology and the traffic flow into a contention graph. The first step in this respect is depicted in Figure 4.2 (top), in which a connection between two nodes indicates that the nodes can sense each other, and a label below a connection reflects an active link. Based on

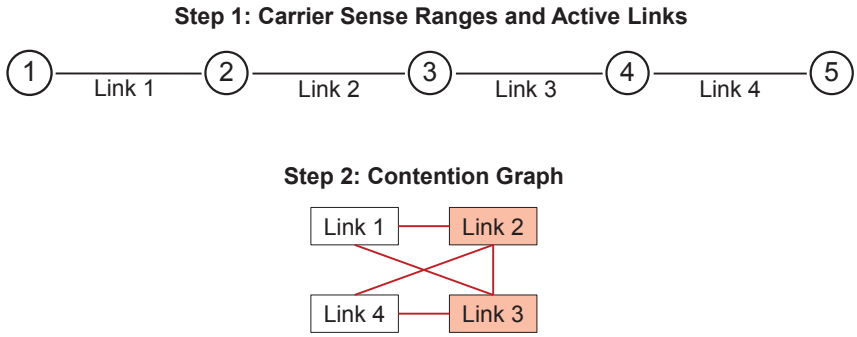

Figure 4.2: Contention graph of the chain network of five nodes.

this information, the contention graph is drawn as shown in Figure 4.2 (bottom). Each vertex in this graph represents an active link, and an edge between two vertices denotes either a contention or an interference between the corresponding links. Two links contend with each other and cannot be active at the same time if the sender of one link is in the carrier sense range of the other. Two links interfere with each other and should not be active at the same time if the receiver of one link is within the carrier sense range of the other. For example Links 1 and 2 are contending links because Node 1 (the sender of Link 1) and Node 2 (the sender of Link 2) are in the carrier sense range of each other. Links 1 and 3 are interfering links because Node 2 (the receiver of Link 1) is in the carrier sense range of Node 3 (the sender of Link 3 and a hidden node for the sender of Link 1). Assuming that all network links have the same link throughput, Figure 4.2 (bottom) shows that Links 2 and 3 are the most connected in the contention graph, thus, they are the bottleneck links in the network as these can only offer 1/4 of the link throughput, i.e., $P_{TX_2} = P_{TX_3} = \frac{1}{4}$. Consequently, given low channel error rates, the maximum sustainable throughput in this network (i.e., the saturation throughput) is approximately $\frac{1}{4} \cdot Thr$. That is it is expected that the chain of five nodes should deliver nearly 100 % PDR in case the traffic load is up to $\frac{1}{4} \cdot Thr$. In case of higher loads, for instance $\frac{1}{3} \cdot Thr$, where only the spatial channel re-use (due to the carrier-sensing mechanism) is considered but not the interference, the PDR should considerably drop. In the following, the static chain of five nodes and the aforementioned throughputs are used as traffic loads to investigate the validity of HWMPS and BATMANS in OMNeT++-INETMANET. That is, using those traffic loads the performance of both protocols is compared in simulation and in practice.

Figure 4.3: Selected photos of the WLAN mesh nodes prototypes.

4.3.2 Reference Testbed for WMNs

To run experimental tests on WMNs and to validate the simulation models used in this thesis, the mesh nodes illustrated in Figure 4.3 (middle) are primarily used. These mainly incorporate the following components:

- A RoBoard RB-110 embedded system [ROB]. This board builds the heart of the units. It is based on a VortexX86 32 bit CPU running at 1000 MHz, and a 256 MB DRAM. It provides a miniPCI slot, two USB ports, and one serial and one Ethernet interfaces. The miniPCI slot is used to integrate the WLAN mesh interface while the USB ports are needed to mount additional wireless cards, e.g., a WLAN access point card on mesh access points or an LTE dongle on mesh gateways. The serial port is used for debugging purposes. The Ethernet interface is used for the configuration and monitoring of the measurements.

- A Wistron DNMA92 miniPCI card, which is integrated in the RB-100 and is used as a WLAN mesh interface. It implements the IEEE 802.11a/b/g/n standards and has support for two bands (2.4 GHz and 5 GHz). It includes

a 2x2 MIMO, thus offering data rates up to 300 Mbit/s in the case of IEEE 802.11n and channel bundling (40 GHz).

- A lithium polymer battery (SLS APL 5000 mAh 2S1P 7.4 V 20C+/40C) for mobility purposes—in addition to the static power supply.

To ease the deployment of specific topologies in different environments, each mesh node has a corresponding tripod, as illustrated in Figure 4.3, in order to appropriately adjust the heights. For UAV-specific measurements, e.g., the air-to-air channel characteristics of UAV-WMN, a flying prototype of the mesh nodes is built [God12b], cf. Figure 4.3 (right).
As for the software configuration of the units, Debian Wheezy based on Linux kernel 3.4.4 is installed. The IEEE 802.11s implementation is included in the kernel and is supported by the ath9k driver used by the DNMA92 WLAN card (backports-3.10.19-1). In addition, the routing protocols BATMAN 0.3.2, BATMAN-adv 2011.1.0 (a MAC-based kernel module implementation of BATMAN), and OLSRd 0.6.1 are installed. For WPA2 pre-shared key, the implementation of IEEE 802.11i personal mode, hostapd-2012.09.10 is deployed. For the IEEE 802.11s security mechanisms, the cozybit authsae-2013.06.05 implementation is used. Various scripts are developed for the semi-automatic configuration, monitoring, and debugging of the measurements, and the Iperf tool is used for traffic generation and performance evaluation.

4.3.3 Performance Evaluation in OMNeT++ and in the Testbed

To validate the OMNeT++-INETMANET models of HWMPS (HWMP + IEEE 802.11s security) and BATMANS (BATMAN + IEEE 802.11i), the performance of both protocols is evaluated in simulation and in practice in the static chain of five nodes, see Figure 4.1. The maximum sustainable throughput, i.e., $\frac{1}{4} \cdot Thr$ (see Sub-section 4.3.1), is offered as traffic load, and both IEEE 802.11 carrier sense mechanisms basic and CTS are considered. Moreover, a traffic load of $\frac{1}{3} \cdot Thr$ in case of the basic carrier sense mechanism is additionally investigated to verify whether the PDR drops, as expected in Sub-section 4.3.1, or whether it remains sustainable. In all measurements IEEE 802.11g is applied, all PHY data rates (for data, control, and broadcast frames) are set to 11 Mbit/s, and a packet size of 1460 Byte is used. Hence, according to Table 4.5, $Thr_{Basic} = 6.92$ Mbit/s and $Thr_{CTS} = 5.10$ Mbit/s (the overhead of IEEE 802.11s/i is negligible). Thereby, the investigated traffic loads are:

- $\frac{1}{4} \cdot 6.92 = 1.57$ Mbit/s, using the the basic carrier sense mechanism.
- $\frac{1}{3} \cdot 6.92 = 2.09$ Mbit/s, using the the basic carrier sense mechanism.
- $\frac{1}{4} \cdot 5.10 = 1.27$ Mbit/s, using the CTS carrier sense mechanism.

Table 4.6: Relevant measurement configurations.

Network configuration

Parameter	**Value**
Carrier sense range [m]	341.8 (simulation)
Interference range [m]	277 (simulation)
Transmission range [m]	267.1 (simulation)
PHY data rate [Mbit/s]	11
Basic PHY data rate [Mbit/s]	11
MAC layer	IEEE 802.11g
Carrier sense mechanism	Basic, CTS
Channel model	Free-space (simulation)
Traffic model	CBR-UDP (Iperf, experimental)
Traffic load [Mbit/s]	1.57 (Basic), 2.09 (Basic), 1.27(CTS)
Packet size [Byte]	1460
Simulation time [s]	300

Routing protocol configuration

Parameter	**Protocol**	**Value**
OGM-Interval	BATMANS	0.5s
RootMode	HWMPS	3 - hybrid
Neighbor-Hold-Time	HWMPS	12s
Route-Hold-Time	BATMANS	5, 200s
Route-Hold-Time	HWMPS	15s

Table 8.4 depicts all relevant configurations. The configuration of the routing protocols is based on the findings in [Sbe14c, Hiy13]. In case of the Route-Hold-Time parameter of BATMANS, an additional value of 200s is considered, even though not relevant for highly dynamic WMN, to validate the impact of different parameterizations on the performance. Fifteen Repetitions for each measurement are performed. The arithmetic mean and 97.5 % confidence interval of the Packet Delivery Ratio (PDR) are depicted in Figure 4.4. The PDR is the ratio of the number of received packets at the destination at the application layer to the number of sent packets by all corresponding sources. Figure 4.4 sheds light on four observations:

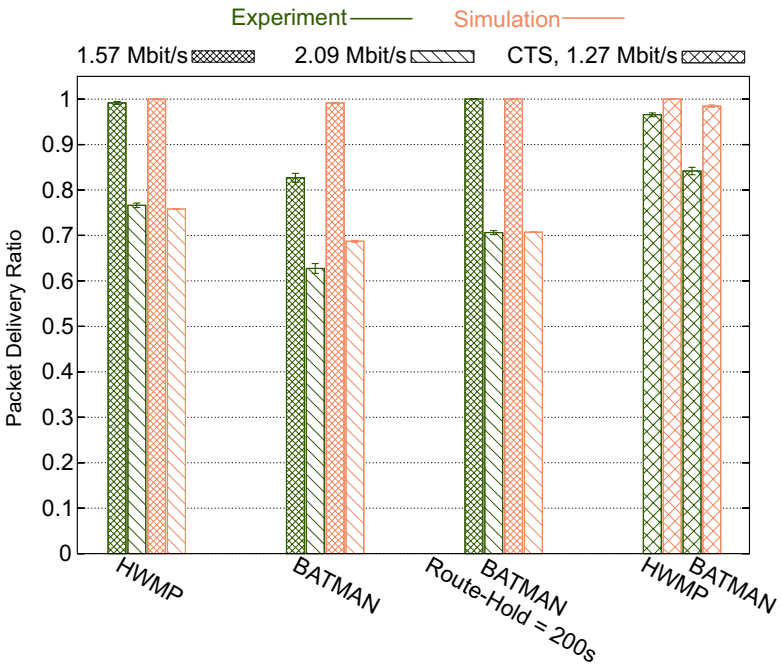

Figure 4.4: Experimental validation of the performance of HWMPS and BATMANS in simulation.

- HWMPS and BATMANS performs very similar in simulation and in practice —in the investigated network. The differences in the PDR results between simulation and practice are mostly below 5 %. This reflects the validity of the OMNeT++-INETMANET models of HWMPS and BATMANS, as well as the validity of the underlying WMN model.
- The results are inline with the theoretical analysis with respect to the maximum sustainable throughput. As Figure 4.4 shows, a PDR of 100 % can be achieved in case the traffic load is 1.57 Mbit/s, and this PDR significantly drops in case the traffic load is 2.09 Mbit/s. The nearly identical PDR drop in simulation and practice points out again the validity of the WMN model in OMNeT++-INETMANET (i.e., channel access, queuing behavior, etc.).
- The relative performance of the protocols is credible. HWMPS performs better than BATMANS in case the Route-Hold-Time of the latter is set to 5 s, as expected. In that case, BATMANS triggers a route timeout every

successive missing of $\frac{Route-Hold-Time}{OGM-Interval} = 10$ OGMs (due to collision or channel errors, see [Sbe14c]). Consequently, the data packets are dropped until the route is restored. In case the Route-Hold-Time parameter is set to 200 s, the route does not get deleted, thus, the PDR increases in this network. A comparison of the performance of HWMPS and BATMANS in case of 2.09 Mbit/s traffic load shows, as expected, that in case the network is over-saturated the message overhead of BATMANS has a non-negligible impact on the performance. That is, Figure 4.4 stresses the validity of the impact of different routing philosophies and parameterizations on the performance in simulation.

- The use of RTS/CTS (1.27 Mbit/s traffic load) does not lead to an increase in the PDR in comparison to the basis carrier sense mechanism (1.57 Mbit/s traffic load). For instance, in case of BATMANS collision-based route breaks still occur (OGM collides with RTS, CTS, and ACK frames, see also [Sbe14c]). That is, the use of RTS/CTS is not effective in multi-hop scenarios as it does not solve the hidden-node problem, and it leads to a decrease in the maximum sustainable throughput. Additional arguments on the ineffectiveness of RTS/CTS can be found in [Cha12, Cha04, Xu03]. Thereby, in the rest of this thesis, only the basic carrier sense mechanism is considered.

The credibility of the simulation methodology applied in this research to evaluate the efficiency of the proposed secure routing approach is shown in this chapter. Additional validation steps of the route discovery time of the evaluated routing protocols are performed in Chapter 8.

5

PASER: Position-Aware, Secure, and Efficient Mesh Routing

Contents of this Chapter

In this chapter, the novel secure reactive routing approach PASER is presented. With PASER, the author strives for a deployable secure routing solution in highly dynamic WMNs in general, in UAV-WMN in particular. Thus, the main goals of PASER are feasibility, efficiency, and resiliency to the relevant routing attacks in the target network.

Publications. Parts of this chapter have been presented in [Sbe15], [Sbe12b], and [Sbe11b].

5.1 PASER Assumptions

The PASER approach assumes the following network and attacker models.

5.1.1 Network Model

As a target network of PASER, A wireless mesh backbone composed of mobile (e.g., UAV) nodes and static (e.g., ground station) nodes is assumed. The network is operated by one organization (e.g., fire brigades), and the network access is restricted. The legitimate operator nodes conform to the system protocols, while malicious nodes might deviate from them. A public key infrastructure is assumed, with the network operator playing the role of the certification authority. The legitimate nodes have a certificate with integrated roles (i.e., gateway, access point, or router). The network operator also runs a secure Key Distribution Center (KDC) that is responsible to dynamically manage network credentials. All nodes know the public key of the KDC. At anytime, mesh gateways (typically the static nodes such as the ground station) can establish a reliable connection to the KDC and vice versa. It is assumed that the legitimate nodes incorporate a positioning device that runs a secure navigation service, e.g., by using the Galileo Public Regulated Service [PRS] or by running a secure position verification scheme, such as in [Lei06]. That is, the target scenario is assumed to be outdoor and to have low obstacles, as in UAV-WMN.

5.1.2 Attacker Model

This research focuses on attacks that target the security of routing. The main objective of the attacker is to manipulate the routes either to sabotage the network or to mount advanced attacks violating the flight security of the UAVs. The attacker is assumed to control a number of external nodes, which might have more power and better communication range than the legitimate nodes. The attacker might also compromise legitimate nodes or network credentials, by means of social engineering, physical attacks, cryptanalysis, and others. Thereby, the attacker can appear in the network as legitimate nodes by utilizing the compromised identities and keys. Nodes under the control of the attacker might arbitrarily deviate from the protocol behavior. In particular, they can drop, modify, or generate corrupted data packets or routing messages. Attacker nodes can also act in coordination, and they can communicate across large distances using additional fast communication channels. The attacker is, however, computationally bounded and cannot break cryptographic primitives. In the following, the most relevant attacks with respect to this attacker model are listed. A distinction is made between external attacks and internal attacks.

- External attacks: The attacker has no access to the network. The attacker can neither generate nor modify routing messages.
  - **Time-based replay**
    Recording routing messages of legitimated nodes and resending these at later times.
    Consequences: Building suboptimal routes or route loops.
  - **Position-based replay**
    Recording the routing messages of legitimate nodes and resending them at another location.
    Consequences: Building suboptimal routes or route loops.
  - **Media Access Control (MAC) address impersonation***
    Faking the identity of legitimate nodes using MAC spoofing. *: This is not a routing attack but it is used as part of a compound routing attack (see the wormhole attack).
    Consequences: Successful processing of frames forwarded by the attacker as legitimate frames in the case of wireless link encryption.
  - **Wormhole**
    This is a compound attack that combines *position-based replay* & *MAC impersonation*. A pair of attackers, linked via a fast transmission path (tunnel), forward routing messages between two distant nodes, making them believe they are direct neighbors.
    Consequences: Building routes that go through the attacker. This leads to selective dropping of packets, sabotage of the network, and flight security violation.
- Internal attacks: The attacker has access to the network. The attacks are mounted after compromising network or node credentials.
  - **Flooding**
    Continuous broadcast of route requests towards non-existing destinations.
    Consequences: Consuming network resources such as bandwidth leading in worst case to denial of service.
  - **Path diversion**
    Forging routing messages generated by legitimate nodes (e.g., tempering the metric).
    Consequences: Building suboptimal routes or route loops.
  - **Fabrication (reactive routing only)**
    Generating forged route replies upon the receipt of route requests; pretending to have optimal routes to destinations.

Consequences: Redirecting and eavesdropping on the corresponding traffic. This leads to selective dropping of packets.

- **IP address impersonation***
  Faking the identity of legitimate nodes using IP spoofing. *: This is not a routing attack but it is used as part of a compound routing attack (see the blackhole attack).
  Consequences: Generating and propagating corrupted information on behalf of other nodes.

- **Blackhole**
  This is a compound attack that combines *path diversion* & *IP impersonation*. The attacker impersonates main destinations in the network (e.g., mesh gateways) and propagates routing messages with higher sequence numbers and better metrics than the original destinations.
  Consequences: Redirecting and eavesdropping on the traffic as well as generating and propagating corrupted information. This leads to selective dropping of packets, sabotage of the network, flight security violation, and UAV hijacking.

5.2 PASER Secure Routing Goals

According to the IETF KARP group, it is not only important to provide secure routing proposals but also to deliver deployable solutions (feasible and efficient), see [Leb12]. Since the target network of PASER is a closed network, PASER aims to meet the following three objectives in the order of priority:

1. Combating external attacks;
2. Dynamically excluding malicious nodes from the network;
3. Minimizing the harm of internal attackers until these have been excluded from the network.

In other words, PASER seeks for establishing and maintaining accurate routes between legitimate nodes. External malicious nodes should not be able to manipulate the routing process or join the network. In case of nodes or key compromise, these nodes should be rapidly excluded from the network, and the keys should be dynamically refreshed. The detection of malicious behavior is not a part of PASER. To implement this goal, a centralized approach running at the ground station might be used. In this way, malicious behavior can be detected based on the deviation of a UAV from its expected behavior and on the anomaly in its key performance indicators. Decentralized detection approaches exist as well, such as in [Bök14]. Adopting the honeypot approach [Pro07], used in the botnet field,

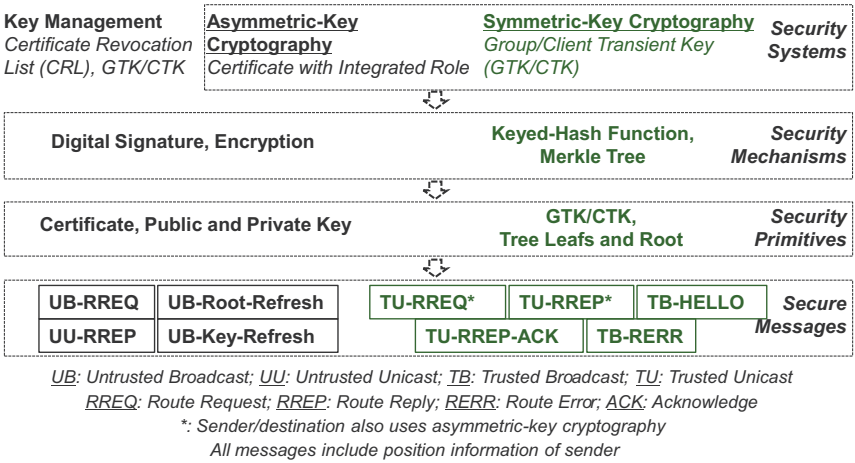

Figure 5.1: Overview of PASER's hybrid security scheme.

could be also an attractive solution.
To achieve its secure routing objectives, PASER seeks to fulfill the following security goals, which are cryptographic goals and can thus be realized using cryptographic techniques (for detailed information about cryptographic goals and information security goals in general refer to [Men96]):

- **Message authentication:** Assuring the party that receives a message that the party which sent the message is an authorized node, and that the message has not been altered by unauthorized nodes during transport.

- **Message freshness:** Assuring the party that receives a message that the message is fresh. That is, it has not been received before.

- **Neighbor authentication:** Assuring one party the identity of a second party involved, and that the second is located in its transmission range.

- **Origin authentication:** Assuring the party that receives a message the identity of the message originator.

- **Dynamic key management:** Providing a dynamic method to distribute and revoke network keys, and to exclude nodes.

5.3 PASER Building Blocks

The PASER approach is composed of three main operations, namely, *node registration*, *route discovery*, and *route maintenance*. These are based on the PASER hybrid scheme and messages, which are illustrated in Figure 5.1. They are defined as follows (see [Sbe12c] for a detailed description).

- *Node registration*: The registration process takes place during the network setup and in case of a key refresh. Hereby, all nodes have to contact the KDC in order to receive the appropriate network keys. A state diagram of this process is illustrated in Figure 5.2. As the diagram shows, mesh gateway nodes directly contact the KDC because these have a reliable access to it. Other nodes (i.e., routers and access points) first have to discover a route towards a gateway in order to reach the KDC. As the diagram illustrates, after finishing the registration process, a node (mesh gateway, router, and access point) possesses the required symmetric network keys to successfully operate in the network. Thus, the node does not need to contact the KDC anymore to authenticate registered nodes moving in its transmission range, as opposed to the IEEE 802.11s/i enterprise mode, which involves the server in every authentication.

- *Route discovery*: The route discovery process in PASER has been adapted from the revised version (v2) of AODV [Per13]. The route discovery takes place when a node wants to send a packet towards a destination for which it does not have a route. An example of this process is illustrated in Figure 5.3. As the figure shows, from a routing point of view, the new node broadcasts a route request in the network. Intermediate nodes, which are already registered and know the route, forward the route request in an unicast way to the destination, they do not reply on behalf of it. Upon receiving the request, the destination sends an unicast reply to the joining node, and the route gets established. From a security point of view, new non-trusted one-hop neighbors (e.g., S-W and S-Z) use the PASER asymmetric scheme to secure the messages and to establish a trust relationship (i.e, untrusted messages secured with asymmetric-key-based cryptographic algorithms), while trusted one-hop neighbors (e.g., all other nodes but S) mainly use the PASER symmetric scheme (i.e, trusted messages secured with symmetric-key-based cryptographic algorithms). A state diagram of the route discovery process is illustrated in Figure 5.4.

- *Route maintenance*: This process has been adapted from the NeighborHood Discovery Protocol (NHPS) [Cla11]. It is illustrated in Figure 5.5. A node detects a broken link in the following two cases (apart from specific timeouts):

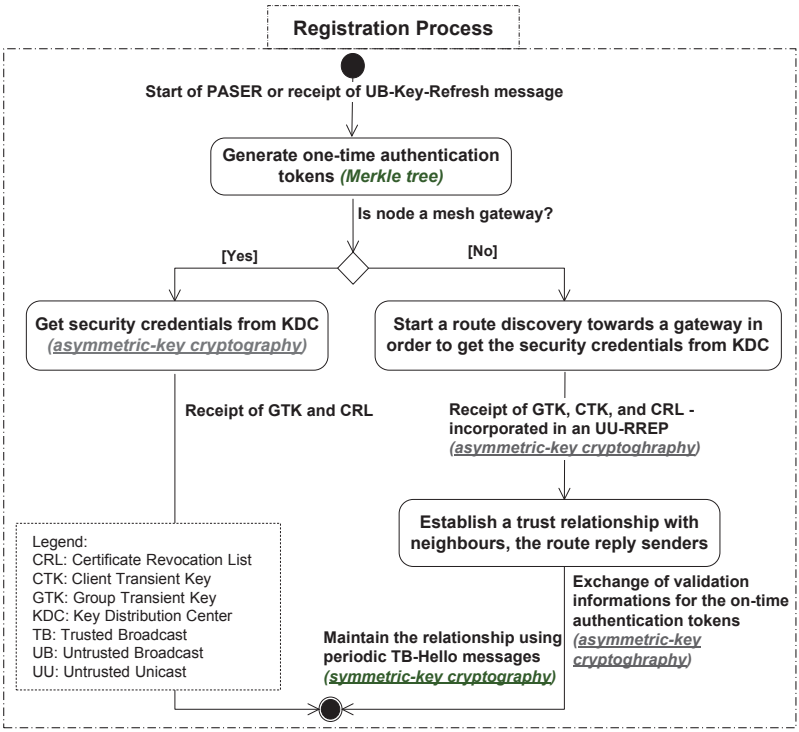

Figure 5.2: State diagram of PASER's registration process.

1. If it has not received a predefined number of trusted broadcast Hello messages from a next-hop for a given route. TB-Hello messages are periodically exchanged between one-hop neighbors. They allow the detection of route changes even in case of no data transfer. Besides, TB-Hello messages enable a proactive discovery of two-hop neighbors since they incorporate a neighbor list. TB-Hello messages also include position information enabling a permanent update of one-hop neighbor positions, which is necessary for protection against the wormhole attack.

2. If the node did not get an acknowledge for an unicast packet sent to a one-hop neighbor, even after a predefined number of retransmissions (typically seven retransmissions). This mechanism enables a fast reaction of PASER on route breaks in case of active data transfer.

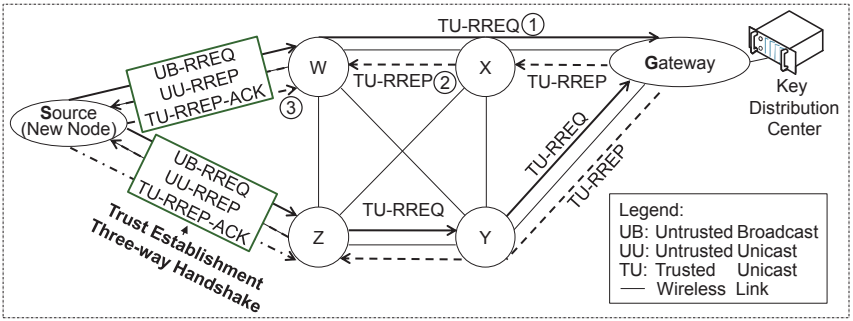

Figure 5.3: Example of the route discovery in PASER during the registration of a new mesh router or access point [Sbe12b].

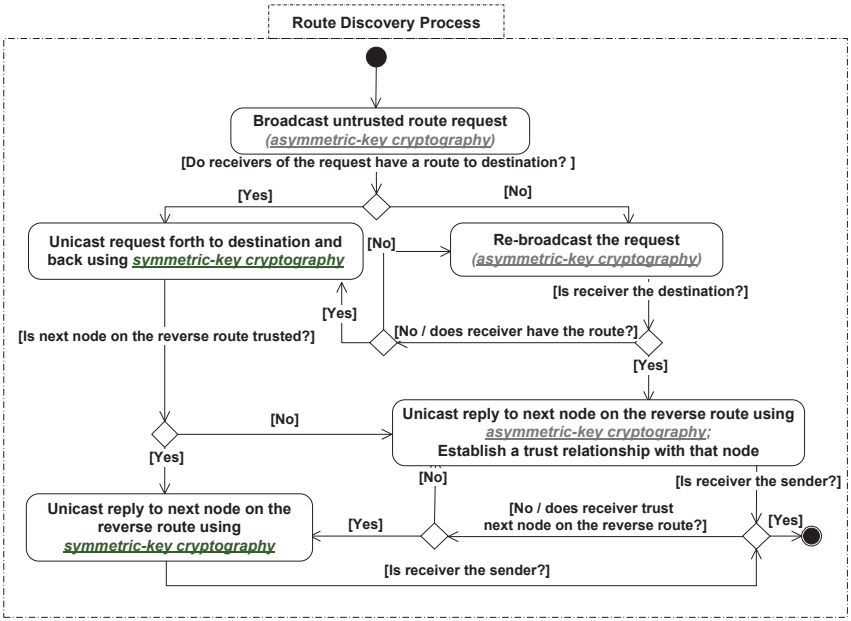

Figure 5.4: State diagram of PASER's route discovery process.

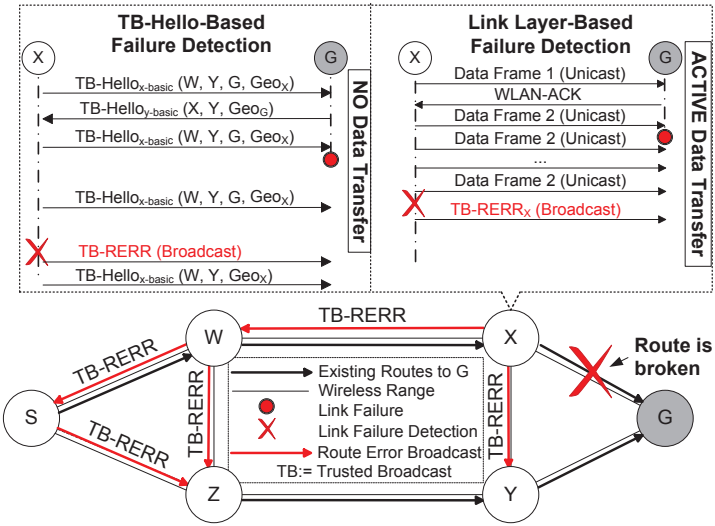

Figure 5.5: PASER's route maintenance approach.

Next, the main security steps in PASER are elaborated. In each step, it is indicated which of the PASER secure routing goals, defined in Section 5.2, are fulfilled and how.

5.3.1 Generation of One-time Authentication Secrets

As a preliminary security step before joining the network, a node generates 2^n pseudo random secrets with $n \in \mathbb{N}$. The construction of such a secret is depicted in Figure 5.6 (bottom). It is l bit long, with $l \in \mathbb{N}$ and $l > n$. The least significant $(l - n)$ bits are random. The most significant n bits constitute an initialization vector. The initialization vector is incremented by one for each subsequent secret. After generating the secrets, the node computes the root element of a Merkle tree [Mer79] using the secrets as leaf pre-images, as depicted in Figure 5.6 (top). The secrets are later included (revealed) in PASER messages exchanged between trusted neighbors to provide **origin authentication**. A secret is only used once to ensure **message freshness** and to prevent time-based replay attacks. The root element is used to validate a secret. It is exchanged during the trust establishment between neighbors.

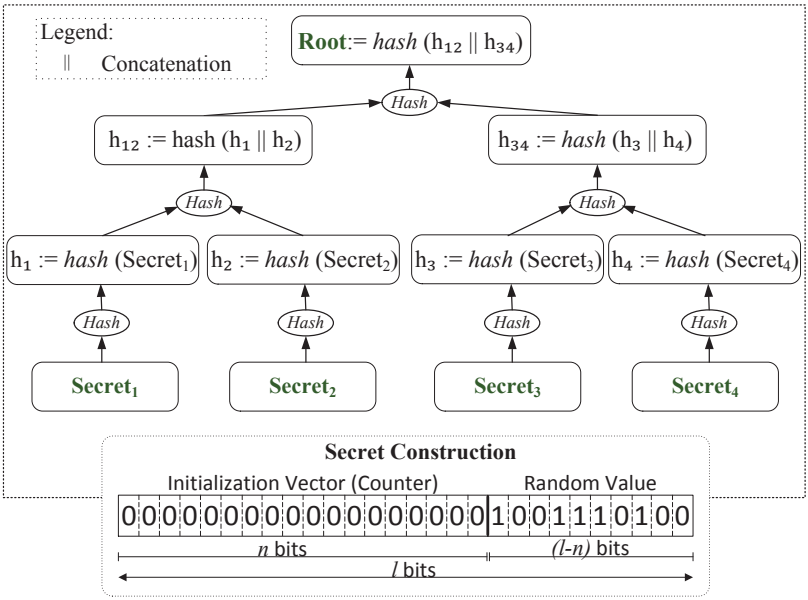

Figure 5.6: Construction of the one-time authentication secrets and the corresponding Merkle tree in PASER [Sbe11b].

5.3.2 Registration of Mesh Nodes

The *node registration* process takes place when a node wants to join the network or in case of a key refresh. Gateway nodes directly contact the KDC because these have a reliable access to it. The process of a gateway's registration is illustrated in Figure 5.7 (top, left). It is based on the PASER asymmetric scheme. As the figure shows, the gateway sends a signed key request, including a nonce and its certificate. After verifying the freshness of the nonce, public key, and signature of the gateway, the KDC replies with a signed key reply that includes the gateway's nonce, a certificate revocation list to report on compromised nodes, a signed ID of the group key in use, and the symmetric group key encrypted with the public key of the gateway. As a result, the gateway possesses the secret key required to use the PASER symmetric scheme, and its is aware of the compromised nodes. The following security goals are achieved during the registration of a gateway:

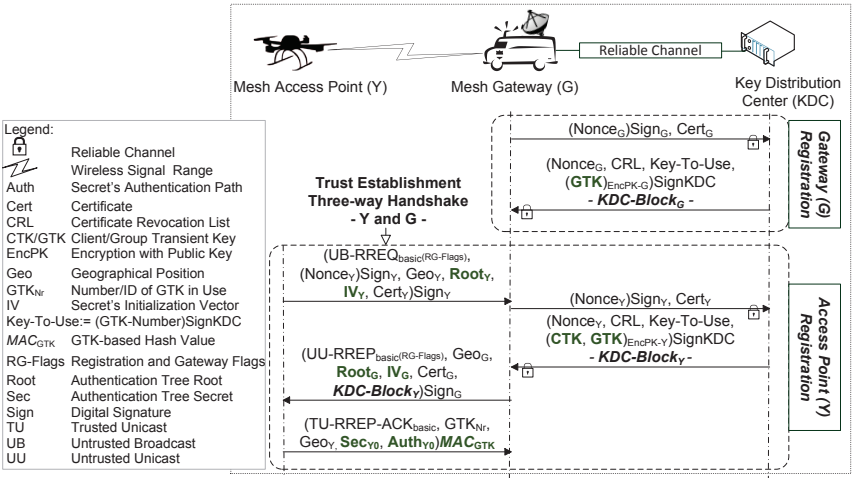

Figure 5.7: PASER's registration process of the mesh nodes at the KDC [Sbe15].

- **Message authentication, and origin authentication**: Due to the use of digital signature;
- **Message freshness**: Due to the use of nonce;
- **Dynamic key management**: Due to the distribution of the encrypted symmetric Group Network Key (GTK) as well as the Certificate Revocation List (CRL).

The *node registration* process of a mesh router or a mesh access point is integrated in the *route discovery* process as these nodes first have to find a route towards a gateway in order to contact the KDC. This process is illustrated in Figure 5.7 (bottom). In this example, a mesh access point Y is one-hop far from a mesh gateway G. As the figure shows, Y broadcasts a signed route request towards a gateway announcing that it wants to contact the KDC (by setting a specific flag). The route request of Y includes a nonce, its position, the root element of its Merkle tree, the current value of the secret's initialization vector reporting which secrets haven been already used, and its certificate. After verifying the distance to Y, its public key, and its signature, G forwards the registration request to the KDC, which replies analogously as during the gateway's registration but with an additional key, the client key. This key is used by the mesh access point to restrict the network access to the authorized clients. G forwards the encrypted keys to Y. It also sends its position, its root element, its IV, and its certificate,

and it signs the whole message. Upon receiving and verifying the message, Y trusts G, and it confirms the receipt of the keys by sending an acknowledgment to G secured with the GTK, among others.
Let originator be the node that generates a message and sender be the node that sends the message, the following security goals are achieved during the registration process of a mesh router or a mesh access point:

- **Message authentication**: Due to the digital signature of the sender;
- **Message freshness**: Due to the nonce of the originator;
- **Neighbor authentication**: Due to the digital signature and position information of the sender. Only those who are located in the carrier sense range are considered as one-hop neighbors;
- **Origin authentication**: Due to the digital signature of the originator applied on non-changeable fields in the message, e.g., the nonce;
- **Dynamic key management**: Due to the distribution of encrypted symmetric network keys (GTK/CTK) as well as the certificate revocation list (CRL).

5.3.3 Secure Communication Between Non-Trusted Neighbors

To secure routing messages between non-trusted registered one-hop neighbors, asymmetric-key cryptography is mainly used, and a three-way handshake is run, after which the one-hop neighbors trust each other. Figure 5.8 (left) illustrates an example of such a communication between a mesh node S and a mesh node Y. In this example, S is a registered node that has just moved in the proximity of Y, and it is looking for a route towards G. As the figure shows, the three-way handshake between S and Y is similar to that of Y and G during the registration step, but the KDC is not involved, and the number of the GTK is included in each message as both nodes need to ensure that they are using the same key.
During this step, the following security goals are fulfilled (for the same reasons as in the registration step of a mesh router or a mesh access point): **Message authentication**, **message freshness**, **neighbor authentication**, and **origin authentication**.

5.3.4 Secure Communication Between Trusted Neighbors

To secure routing messages between trusted one-hop neighbors (e.g., Y and G after the registration), the PASER symmetric scheme is applied, as illustrated in the route discovery example of Figure 5.8 (right). An overview of this step is depicted in Figure 5.9. Here, the following security goals are achieved:

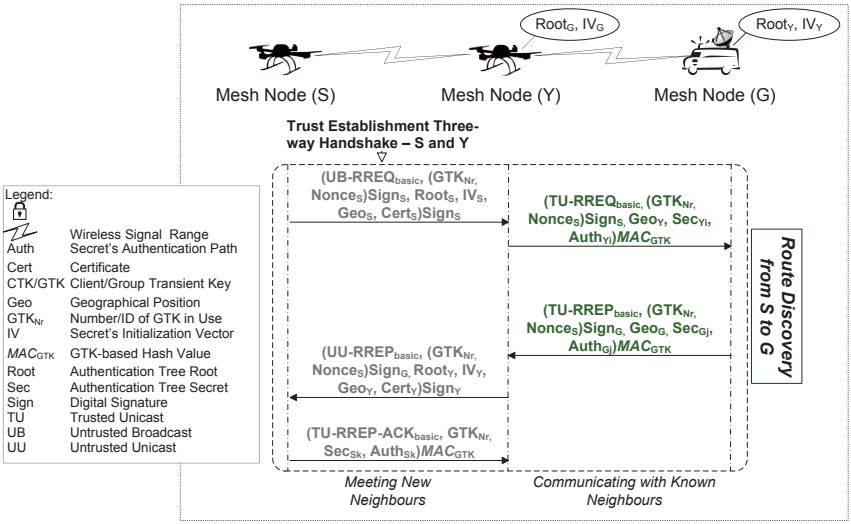

Figure 5.8: PASER's trust establishment between new neighbors.

- **Message authentication**: Due to a keyed hash function MAC based on the group transient key, which is only known to the authorized nodes.

- **Message freshness & neighbor authentication**: Due to the one-time authentication secrets. A sending node (Y) discloses a secret (e.g., $Secret_1$ in Figure 5.1), and sends it along with the corresponding authentication path (e.g., h_2 and h_{34}) and the routing message to the next-hop (G), as illustrated in Figure 5.9. The authentication path of a secret consists of all siblings of the secret's corresponding leaf on the path between that leaf and the root. To verify the disclosed secret, a receiver needs to compute the potential values of the secret's ancestors by iteratively using of the hash function. A secret is authenticated and accepted as correct if and only if the computed root value is equal to the already known root value of the node. That is, neighbor G, already knowing the IV and root element of Y, verifies if the secret is fresh ($IV_{Secret_1} > IV_Y$). It computes the root of the secret it has received and verifies if it matches the root of Y ($Root(Secret_1, h_2, h_{34}) = Root_Y$). If true, G can trust that the message has been sent by Y. Consequently, G ensures that the message has been sent by a one-hop neighbor since G only possess the root elements of one-hop neighbors.

Figure 5.9: Symmetric scheme for secure communication between trusted one-hop neighbors in PASER [Sbe15].

- **Origin authentication**: Due to the forwarding of digital signature of the message originator in case of route requests or replies, and due to one-time authentication secrets in case of one-hop messages such as TB-Hello, as the origin is the one-hop neighbor in that case.

5.3.5 Dynamic Key Management

The dynamic distribution of the group/client transient keys (GTK/CTK) during the network setup (i.e., *node registration*) was shown in Figure 5.7. Upon receiving the GTK during the network setup, message originators always include (and sign) the number of the GTK in each PASER message (cf. Figure 5.8 and Figure 5.9). This number is verified at each node that processes the message. In case of a key or node compromise, the certificate of the compromised node is blacklisted, and a key re-generation process is triggered at the KDC, as depicted in Figure 5.10. Consequently, a new Key-To-Use mark, $(GTK_{Nr_{Old}} + 1)Sign_{KDC}$, is flooded in the network. Upon receiving the new mark, each node resets its routing table and re-registers itself at the KDC. The node then receives the new keys and an updated Certificate Revocation List (CRL), i.e., the node gets informed about the compromised node. If a legitimate node has not received the reset message due to interference or channel propagation error, it detects from the higher key number in use that a key refreshment has occurred. Its neighbors even proof that by using the new Key-To-Use mark, originated and signed by the KDC, see Figure 5.10. Due to the Key-To-Use mark, an attacker who compromises a node, cannot denial the service of its neighbors by just increasing the key number of its messages. Besides, due to the digital signature of the KDC and the GTK counter, **message authentication**, **message freshness**, and **origin authentication** are guaranteed during the refreshment of the keys.

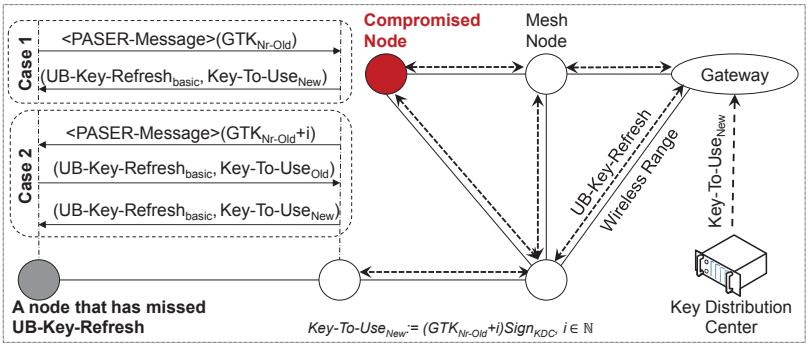

Figure 5.10: Example of a dynamic key refreshment request in PASER.

5.4 Time Costs of the PASER Cryptographic Operations

As a first step towards a thorough performance evaluation of PASER, the time costs of the PASER cryptographic operations are measured on the Roboard RB110 (x86, 1 GHz, 256 MByte DRAM, see Sub-section 4.3.2 for the detailed description of the node). The evaluated asymmetric cryptographic operations are RSA digital signature (using 1500 Byte random input) and RSA encryption/decryption (one operation). The key size used is 1024 bit. Even though published work in the early 2000s estimated that a 1024 bit RSA key can be broken in a year by an attack machine costing significantly less than 10^9 dollars (see [Ber14a] and the references therein), and even though the work in [Ber14a] shows that the attack time and cost can be reduced, this key size is still used in this research for the following reasons: a) RSA is applied in PASER to provide a real-time security of the routing messages. In this context, long term security is not a concern. b) It is striven for minimal time costs of the cryptographic operations.

The evaluated operations of the PASER symmetric scheme are HMAC (using 1500 Byte random input), the generation of the one-time secrets of the Merkle tree, and the verification of a secret. The cryptographic hash function used is SHA-256 (i.e., SHA-2 the successor of SHA-1). The secret size is set to 256 bit with an initialization vector of 32 bit. All operations are implemented using the C language and the OpenSSL library. Ten repetitions for each measurement are performed using ftrace. The average time cost of HMAC is 0.141 ms. The average time costs of the remaining operations are depicted in Figure 5.11 (the confidence intervals are omitted as the standard deviations are negligible).

Figure 5.11 (left) shows the time costs of the PASER asymmetric operations.

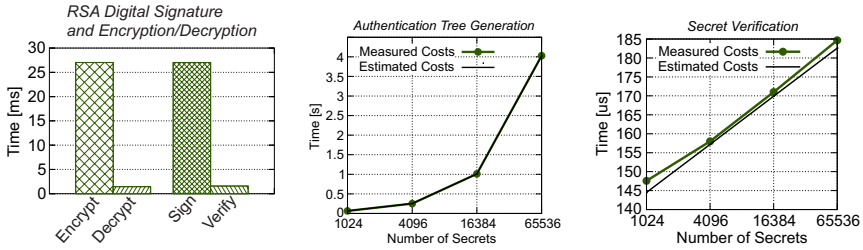

Figure 5.11: Avgerage time costs of PASER's cryptographic operations (experimental) [Sbe12b].

The figure depicts that decrypting and signing have by far a higher cost than encryption and verification (approximately 27 ms). This is caused by the length of the private exponent/key used in decryption and signature, which needs to have a similar length as the RSA modulus (1024 bit). In contrast, the public key used in encryption and verification is very short, e.g., 17 bit. This makes the use of the expensive operations, decryption and signature, for each message by each node inefficient, and it raises the need for a symmetric scheme.
The average time costs of the PASER one-time authentication scheme are provided in Figure 5.11 (middle and right). Let $t_{treeGeneration}$ and $t_{verification}$ denote these costs, i.e., generation of the Merkle tree and secret verification, they can be estimated as follows:

$$\begin{aligned} t_{treenGeneration} &= (n-1)\cdot(t_{sec}) + 2\cdot(n-1)\cdot t_{hash} + t_{sec_i} + t_{hash_i} \\ t_{verification} &= log(n)\cdot t_{hash} + t_{hash_i} \end{aligned} \tag{5.1}$$

Here n is the number of secrets, and t_{sec_i} and t_{hash_i} are the initial costs of a secret generation and a hash operation, respectively. The initial costs occur when the operations' instructions and data have not been loaded in the processor cache yet. The average value of these costs are $t_{hash_i} = 81\ \mu s$ and $t_{sec_i} = 2510\ \mu s$. The average costs of successive calls of these operations are $t_{hash} = 6.35\ \mu s$ and $t_{sec} = 55\ \mu s$.
Figure 5.11 (middle and right) depicts the estimated time costs according to Equation 5.1 as well as the real costs on the mesh node. As the figure shows, the equation provides a nearly perfect estimation of the costs. Figure 5.11 (middle) illustrates that the time cost of the generation of the Merkle tree to secure 65536 PASER messages is almost 4 s. This generation, however, occurs only at network setup, or after all the secrets are used up. In the former case, the 4 s are definitely less than the time required to deploy the mesh nodes in UAV-WMN (after turning them on). In the latter case, the regeneration of the tree might occur seamless when the processor is idle. In case of multi-core processors, it can be also run in parallel. That is, these 4 s should be negligible in practice.

A more relevant impact have the results in Figure 5.11 (right). This figure depicts the time of the secret verification operation, which is run upon the receipt of any routing message secured with the PASER symmetric scheme. As the figure shows, this time is very lightweight, it is less than 200 μs by a tree consisting of 65536 secrets. Consequently, the time cost of the PASER symmetric scheme is less than 341 μs in that case. That is, it is more than 75 times faster than the asymmetric scheme, and this is a gain achieved at each hop by each message.

6

Implementation of PASER in Simulation and in Practice

Contents of this Chapter

This chapter covers the modular implementation design of PASER in OMNeT++ and its cross-layer implementation in Linux. Particular highlights are the integration of the PASER protocol in the official INETMANET framework of OMNeT++ and the attestation of the feasibility of PASER in practice.

Publications. Parts of this chapter have been presented in [Sbe13a], [Sbe13b], and [Sbe14b].

6.1 Implementation in INETMANET-OMNeT++

This section gives an overview of the implementation code of PASER in the INETMANET framework of the network simulator OMNeT++. Especially, the modular design of this implementation and splitting OMNeT++ specific code from the rest of the code are noteworthy.

6.1.1 Goals of the PASER Implementation in Simulation

Bearing in mind that the final target beyond the simulation of PASER is its deployment in real life scenarios, the PASER implementation in OMNeT++ have had to fulfill the following goals in the order of priority:

1. Performance evaluation of the protocol and design optimization if necessary;
2. Verification of the protocol robustness against well-known attacks and design optimization if necessary;
3. Providing a portable code to Linux and to other simulation tools than OMNeT++;
4. Contributing a reference easy-to-adapt implementation of a secure routing protocol in simulation.

6.1.2 The Big Picture of the PASER Implementation in Simulation

To achieve the envisaged goals in Sub-section 6.1.1, the PASER C++ code in OMNeT++ is divided into sub-modules as depicted in Figure 6.2, where each sub-module is responsible for a specific task. This modular design allows low-effort protocol optimizations, as only the corresponding modules need to be changed. Moreover, OMNeT++ specific code is mainly kept in the 'Socket' sub-modules, which enables a straightforward porting of the rest of the code to Linux and to other simulation tools than OMNeT++. As Figure 6.2 shows, the PASER code in simulation is mainly composed of:

- PASER Logic
  - **Route Discovery:** It is a set of functions that manage a node registration at the KDC and handle a route discovery.
  - **Route Maintenance:** It provides functions to keep routes up-to-date. It manages several PASER timers and the link layer feedback.
  - **Timer Management:** It manages all PASER timers.
  - **Socket**: it includes the *handleMessage* function that is called upon receipt of in/outcoming messages, i.e., it is the interface to the outside world.
- PASER Data Structures and Support Library

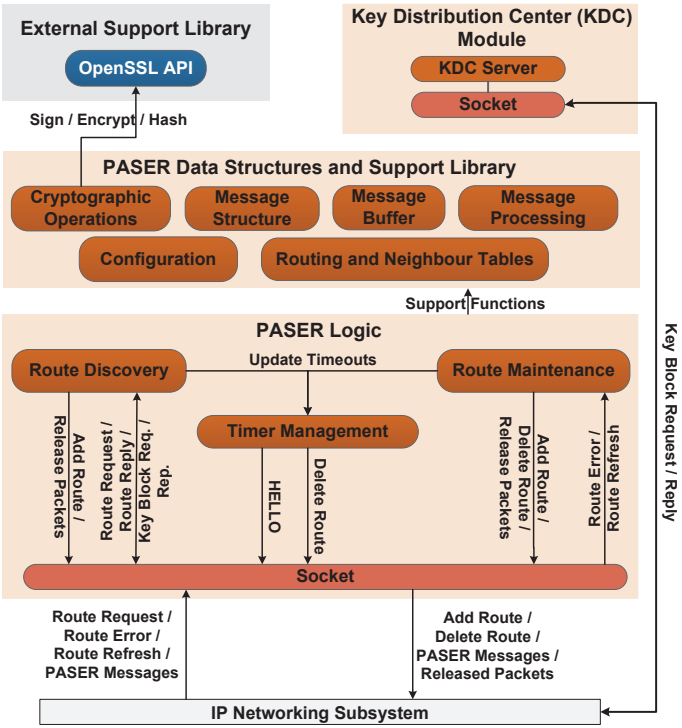

Figure 6.1: The big picture of the PASER implementation in OMNeT++ [Sbe13b].

- **Cryptographic Operations:** It handles all security related operations. This sub-module depends on the external library OpenSSL [SSL].
- **Message Structure:** It comprises the classes of all the PASER messages, which are illustrated in Figure 5.1 (bottom) in Chapter 5.
- **Message Buffer:** It manages a buffer of all data packets that must be forwarded to an unknown destination.
- **Message Processing:** It provides functions to process all PASER messages.
- **Configuration:** It includes a parser of the PASER configuration parameters in the NED file. These include the delays of the PASER

operations in practice, see Figure 5.11 in Chapter 5. It also includes additional PASER configuration parameters for developers.

  - **Routing and Neighbor Tables:** It manages the PASER tables.

- Key Distribution Center Module
  This module handles the key management in PASER. Nodes contact this module in order to get the required keys to access and operate in the network.

The implementation of PASER has been recently integrated in the INETMANET framework as the first implementation of a secure routing protocol in OMNeT++, see [INE13]. This implementation is used in the next chapter to evaluate PASER in various UAV-WMN scenarios.

6.2 Implementation in Linux

The PASER implementation in Linux is cross-layer, and it is composed of the routing logic in the Linux user space and a novel generic framework in the Linux kernel space, called ROUTE-O-MATIC. Figure 6.2 gives a big picture of the PASER C/C++ language-based implementation in Linux.

6.2.1 Routing Logic

The heart of the routing logic is the **Scheduler** (see Figure 6.2, middle). It is mainly responsible for the following tasks:

- Monitoring incoming messages from the PASER kernel module at the **Netlink Client**.
- Watching incoming messages from PASER peer nodes at the **PASER Gate** (UDP socket).
- Monitoring the replies of a key distribution center at the **KDC Client** (TCP socket).
- Watching the **Timer Management**, waiting for route timeouts to occur.
- Triggering route request for unknown destinations at the **Route Discovery** component.
- Ordering the **Route Maintenance** component to delete or refresh existing routes.

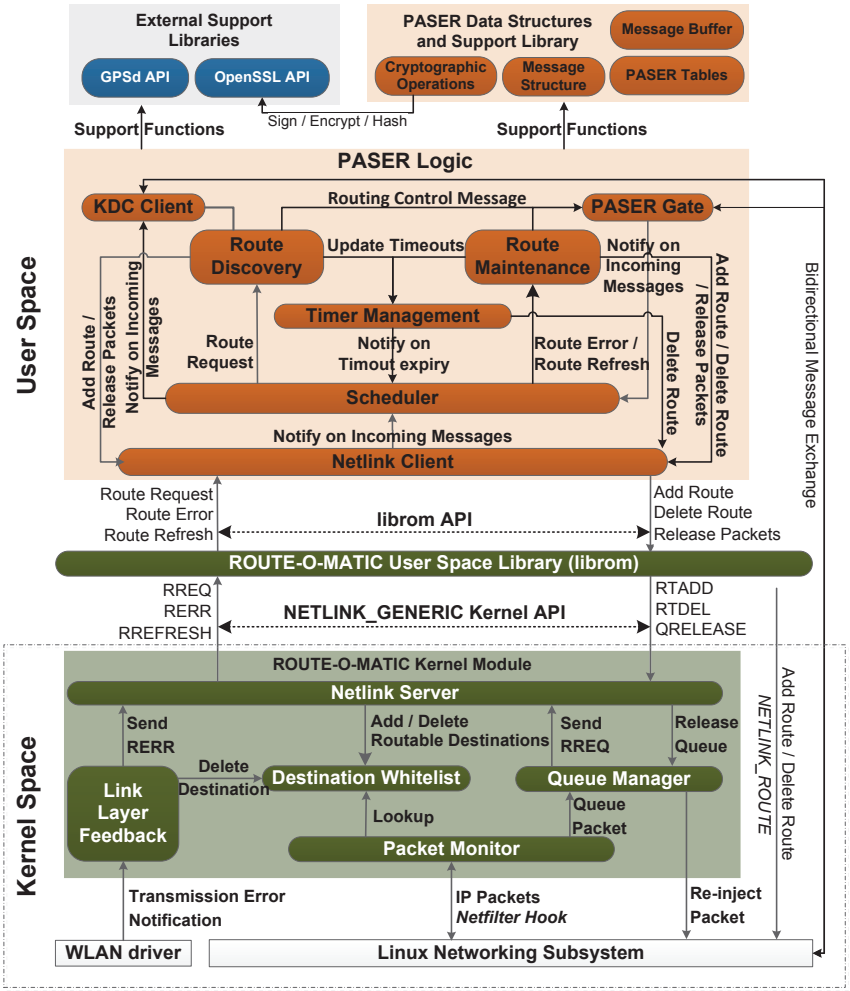

Figure 6.2: Big picture of the PASER implementation in Linux.

The protocol logic relies on a PASER library, which offers support for messages' processing, and maintenance of routing and neighbor tables. The PASER logic also depends on the OpenSSL library [SSL] to perform its cryptographic operations. It also uses the GPSd Application Programming Interface (API) [GPS] to read the geographic positions of nodes.

While most of the routing logic components were ported from the PASER code in OMNeT++ (see Section 6.1), the main challenge in implementing PASER in Linux was the development of a kernel module, for which deep knowledge of Linux network internals were required. Here, the novel ROUTE-O-MATIC *framework has been proposed, which is not specific to PASER, but it is designed to be used by any IP-based reactive routing protocol to ease its implementation.*

6.2.2 Generic Kernel Framework: ROUTE-O-MATIC

This section outlines the implementation design and performance evaluation of ROUTE-O-MATIC (see Figure 6.2, bottom), a comprehensive framework for IP-based reactive (mesh) routing in Linux. This framework is used to implement PASER. It provides a set of necessary but not natively supported services and interfaces for the development of IP-based reactive (mesh) routing protocols in Linux.

6.2.2.1 The Need for route-o-matic

Reactive routing logic such as that of PASER mainly relies on two core features of the underlying operating system:

1. A mechanism which notifies the routing logic in case a route to an unknown destination is needed;
2. A buffer which temporarily saves the packets of the unknown destination while the route discovery is performed and re-injects those after establishing the route.

The network subsystem of the Linux operating system, which is the default platform for the implementation of wireless mesh networks, lacks of support for both features.
Kawadia et al. discussed the challenges of developing IP-based reactive MANET routing protocols in [Kaw03]. They presented a framework implementation termed Ad-hoc Support Library (ASL). Their framework consists of a shared user-space library and a kernel module for Linux 2.4. Several routing protocols such as AODV-UIUC were implemented using ASL. Unfortunately, ASL is no longer supported in up-to-date Linux kernels (newest version is $> 3.4.4$). Chakeres et al. also examined in [Cha05a] different design strategies of AODV implementation on older Linux. Well-maintained kernel components used in their approach, e.g., *Netfilter* hook, have been adopted in ROUTE-O-MATIC. Other components got outdated. Thorup gives in [Tho07] an in-depth presentation of the design, implementation, and evaluation of DYMO-AU. His work has in common with the previous approaches that both the routing logic and the decision maker,

e.g., whether to start a route discovery or not, are mostly implemented in user space. That is, all incoming/outgoing packets must be processed in user space, causing a huge number of unnecessary transitions between kernel and user space. Apart from that, the aforementioned implementations used different components and interfaces to realize the two core routing features. The lack of a common interface led to the current situation, where all these implementations do not run on the recent Linux kernel versions. This has burdened the development of new reactive protocols, during which the same challenges must be faced again, and the developer is required to have a deep understanding of the system's network internals, which increases the implementation and maintenance complexity.
To ease the development of IP-based reactive (mesh) routing protocols in Linux, ROUTE-O-MATIC is proposed in this thesis. It provides an open source generic framework and a simple API for reactive routing logics. The framework is based on well-maintained core kernel components. Its main novelty lies in the simplicity of its design while providing all required functionalities based on an accurate exploitation and an efficient interconnection of relevant kernel components. ROUTE-O-MATIC retains the decision maker in the Linux kernel while offering a lightweight notification for user space routing logics. Additionally, the framework incorporates a link layer feedback support, which reports every transmission failure in order to accelerate the detection of broken links.

6.2.2.2 Design Space Exploration of route-o-matic

ROUTE-O-MATIC is meant to process every IP packet, it has to detect broken links, and must interact with the routing logic in time critical applications. Thereby, the most important goal by the design of ROUTE-O-MATIC is to keep its delay minimal. Figure 6.2 (bottom) shows in a dark green color all components of ROUTE-O-MATIC, including their internal and external interaction. These components are designed as follows.

- **Destination Whitelist:** ROUTE-O-MATIC's decision on whether a packet needs to be queued or not is based on a single bit of information per destination, namely, *if a valid route exists or not*. This information is returned by looking up that destination in the Destination Whitelist. The latter is implemented as a flat table holding all destinations with valid routes. No further information about the route itself is stored in that list, such as which egress device or intermediate node (next-hop) to use. The forwarding part is left to the kernel. In real-world use cases, a mesh router's route table typically consists of a handful of entries: a couple of neighbor routers in the same subnet and one default gateway (i.e., < 128 entries). With these dimensions in mind, the corresponding Whitelist-lookup complexity of $O(n)$ is low.

- **Packet Monitor:** The Packet Monitor is responsible for examining all network traffic. For each IP packet, its destination address is looked up in the Destination Whitelist. In case of a match, the packet is handed over to the kernel for further processing. Otherwise, the packet is forwarded to the Queue Manager which is among others responsible for notifying the user space library *librom*, i.e., the routing logic about the unknown destination. Several traffic monitoring strategies exist for Linux [Cha05a]. ROUTE-O-MATIC uses the kernel *Netfilter* hooks for this purpose. In contrast to the other strategies, *Netfilter* provides a good trade-off between performance, maintenance, and portability.

- **Queue Manager:** A core feature of ROUTE-O-MATIC is the ability to buffer packets until a route to their destination is established—instead of directly rejecting a connection attempt. *Netfilter* provides for this purpose the *ip_queue* service, which stores the packets in user space. However, copying each packet to user space just for buffering and re-injecting them afterwards to kernel space creates an expensive processing overhead. Since ROUTE-O-MATIC is designed to work on resource constrained embedded systems, and the time costs of packet processing are very critical, packet buffering using this framework should solely occurs inside the kernel. Hence, the Queue Manager adopts the built-in kernel data structure *kfifo*, which has also been used in DYMO-AU [Tho07].

  Upon receiving the first packet for an unknown destination, the Queue Manager is first responsible to trigger a RREQ (*Netlink* route request message), which is sent to the user space routing logic. Afterwards, a distinction is made between private and public traffic as follows:

  – If the unknown destination is a private address, a separate queue is created to hold all received packets for that destination. After finding the route, all corresponding buffered packets are re-injected to the network subsystem in order to proceed their transmission. The queue buffer is used in a non-persistent way. That is, it is created when the first packet is received and destroyed after all packets are released. To efficiently manage several queues, these are organized in a double-linked list. To minimize the size of queues, a queue only holds references to the buffered packets. The memory management of those packets is handled by the kernel.

  – In contrast to private network traffic, all packets with public destinations share one queue. They all depend on the default gateway route, and once the latter is established, all of them get released. This shared queue is designed to be persistent due to the high probability of its utilization. Thus, this queue is created upon loading the ROUTE-O-MATIC module, and it remains available until unloading the ROUTE-O-MATIC.

Table 6.1: Selected message types of the ROUTE-O-MATIC *Netlink* protocol [Sbe13a].

ROUTE-O-MATIC → Routing Logic

Message Type	**Action**
RREQ	Request a route to a specific destination
RERR	Report a broken route

Routing Logic → ROUTE-O-MATIC

Message Type	**Action**
RTADD	Add an entry to the Destination Whitelist and release all queued packets for a specific destination
RTDEL	Delete an entry from the Destination Whitelist
RTDMP	Print out the Destination Whitelist
SETGW	Set the default gateway route as valid
QREL	Release queued packets for a specific destination
QDMP	Show the current queues and their capacities

- **Netlink Server:** Linux provides several mechanisms for the communication between the kernel and user space, e.g., *system calls*, *ioctl*, *shared memory* or the socket based *Netlink* protocol family [JS03]. In ROUTE-O-MATIC, the latter one has been used for two reasons. First, *Netlink* does not only support data transfer in both directions, but it also allows both sides to initiate a communication session. This feature is necessary to send RREQ notifications from the kernel to user space, instead of implementing inefficient polling mechanisms. Second, *Netlink* provides a certain degree of flexibility with its concept of user-defined commands and attributes. *Netlink* commands represent message types while the attributes correspond to the payload. With respect to ROUTE-O-MATIC, a command could be a RREQ and its attribute would be the requested IP address. Table 6.1 specifies selected supported message types in current ROUTE-O-MATIC version. Using these messages, the user space component *librom*, i.e., the routing logic can access all the services the kernel part offers.
  The *Netlink* protocol family only supports 32 protocols, among others, *Generic Netlink* [GEN09]. To circumvent this limitation, the latter serves as a general purpose multiplexer. ROUTE-O-MATIC uses *Generic Netlink* to avoid conflict with any officially registered *Netlink* protocol in the future.

- **Link Layer Feedback:** To accelerate the detection of broken links/routes, ROUTE-O-MATIC offers a Link Layer Feedback (LLF) to a routing logic. Without this service, reactive routing protocols usually use periodic hello messages with a typical interval of 1 s to keep the neighbor links up-to-date. That is, in worst case, they need more than 1 s to detect a broken link. By using the LLF, however, the driver immediately reports each failed transmission of a frame. Nonetheless, in the used Linux kernel version 3.4.4, there is no generic driver-independent interface to subscribe for such a transmission error notification. Thereby, in this version of ROUTE-O-MATIC, the Atheros *ath9k* WLAN driver was extended by only a few code lines as a temporarily solution to offer this feature. Choosing a specific WLAN device driver generally contradicts with the portability goal. Nevertheless, selecting Atheros as a reference, which is a common driver for most embedded devices' WLAN interfaces, should minimize this portability issue. Since many different wireless activities around the Linux kernel have been consolidated on the Linux Wireless site [LIN], the authors also expect significant development towards an official, driver-independent LLF interface. The extension of the WLAN driver allows ROUTE-O-MATIC to register a callback function, which is invoked every time a frame transmission for a given destination fails. In that case, the LLF component of ROUTE-O-MATIC deletes this address from the Destination Whitelist and sends a RERR (*Netlink* route error message) to the user space.
- **route-o-matic User Space Library (*librom*):** It provides an easy-to-use API to a user space routing logic. *librom* endorses a set of simple functions that meets the basic need of a routing logic. It handles all *Netlink* communication with the underlying system. It depends on the user-friendly library *libnl* [LIB]. It uses the *NETLINK_ROUTE* protocol to directly access the Linux routing subsystem and to maintain its routing table. This makes *librom* as simple to use as other C libraries.

6.2.3 Performance Evaluation of ROUTE-O-MATIC

The ROUTE-O-MATIC implementation is evaluated in two steps: First, its main functionality is validated as a proof of concept. Second, the performance of ROUTE-O-MATIC with respect to delay/runtime is investigated. Here, the kernel processing time without ROUTE-O-MATIC is first measured as a reference. Afterwards, the runtime of the framework's main operations are subject to several evaluation scenarios.

Proof of Concept and Evaluation Setup. All tests and evaluations were performed using the reference mesh unit Roboard RB110 (see Sub-section 4.3.2).

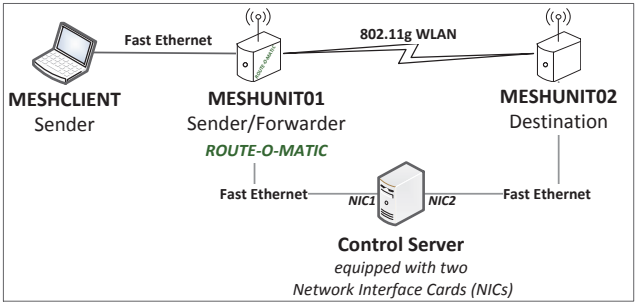

Figure 6.3: Network setup for the proof of concept and evaluation of ROUTE-O-MATIC.

Two mesh nodes were connected via an IEEE 802.11g wireless ad-hoc network, as illustrated in Figure 6.3. The wireless network was dedicated to the measurements, i.e., it was not used for any control purposes. To control the mesh units, a wired Fast Ethernet link was used. Here, each mesh unit was connected to a separate subnetwork. One mesh unit, MESHUNIT01, was equipped with ROUTE-O-MATIC, and it was subject to all measurements. The second mesh unit, MESHUNIT02, was used as a remote station replying to ICMP echo requests as well as acting as *iperf* server. A third node, MESHCLIENT, was used to evaluate the forwarding mechanisms of ROUTE-O-MATIC. *ftrace* was used for a high runtime precision analysis level. It is a built-in kernel debugging feature which allows to trace every function call with minimal overhead [Bir09]. On top of that, the preemption model of the operating system was deactivated (no forced preemptions) to keep the number of context switches minimal.

6.2.3.1 Proof of Concept

The general functionality of ROUTE-O-MATIC was validated by locally sending generated packets as well as by forwarding incoming packets.
First, the average Round-Trip Time (RTT) for hundert ICMP packets from MESHUNIT01 to MESHUNIT02 was determined using *ping*, without ROUTE-O-MATIC being loaded. This average RTT of 1.61 ms was used as a general reference time. Afterwards, ROUTE-O-MATIC was loaded on MESHUNIT01 and the previous measurement was repeated. Figure 6.4 depicts the output of *ping* during this measurement. The figure sheds light on the expected queuing effect of ROUTE-O-MATIC. After loading the framework between the 6th and 7th packet, and since the Destination Whitelist was still empty, all the packets got queued until the route to the destination was found. This route entry was manually triggered after 5 s, which

```
MESHUNIT01:~# ping MESHUNIT02
[...]
64 bytes from 192.168.33.1: icmp_req=5 ttl=64 time=1.94 ms
64 bytes from 192.168.33.1: icmp_req=6 ttl=64 time=1.57 ms
64 bytes from 192.168.33.1: icmp_req=7 ttl=64 time=4906 ms
64 bytes from 192.168.33.1: icmp_req=8 ttl=64 time=3896 ms
64 bytes from 192.168.33.1: icmp_req=9 ttl=64 time=2896 ms
64 bytes from 192.168.33.1: icmp_req=10 ttl=64 time=1896 ms
64 bytes from 192.168.33.1: icmp_req=11 ttl=64 time=897 ms
64 bytes from 192.168.33.1: icmp_req=12 ttl=64 time=1.79 ms
64 bytes from 192.168.33.1: icmp_req=13 ttl=64 time=1.59 ms
[...]
```

Figure 6.4: Proof of concept of the queue functionality [Sbe13a].

justifies the RTT increase from 1.57 ms to 4906 ms by the 7th packet. That is, this delay was the time the packet spent in the queue.
The aforementioned procedure was repeated for forwarded packets instead of locally generated ones. In this case, MESHUNIT01 received packets from MESH-CLIENT destined to MESHUNIT02. Both tests in this scenario (i.e., without and with ROUTE-O-MATIC) behaved analogous to the previous scenario but with slightly larger RTTs caused by the extra hop delay. In other words, with respect to ROUTE-O-MATIC, it does not make a difference whether the mesh node running this framework is a source or a forwarder.

6.2.3.2 Scenario I: Reference UDP Packet Processing Time

The evaluation scenarios of ROUTE-O-MATIC are outlined in Table 6.2. The goal of the first scenario was to determine the pure kernel processing time of a UDP packet without the ROUTE-O-MATIC module. This processing time was used afterwards as a reference time to evaluate the overhead of ROUTE-O-MATIC. To measure the pure kernel processing time, the runtime of the kernel function `inet_sendmsg` for locally generated UDP packets was evaluated. This time covers the processing of the packet at the transport, the network, and the link layer, see Figure 6.5.

Results: An average `inet_sendmsg` runtime of 138.89 μs was measured. Compared to the *ping* RTT of 1.61 ms, the kernel processing of a single UDP takes less than 10 %.

Table 6.2: Evaluation scenarios of ROUTE-O-MATIC [Sbe13a].

Scenario	Goal	Parameters	Values
I	**Reference processing time**	#UDP packets	1000
		Data rate [Mbit/s]	2
II	**Route table's lookup delay**	Runs	10 x 10s
		Data rate [Mbit/s]	0.256, 2, 8
		Route table entries	1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048
III	**Queue delay**	Runs	100
		Data rate [Mbit/s]	2
		Queue size [#packets]	1, 256, 512, 1024, 2048, 4096
IV	***Netlink* delay**	Runs	100
		Data rate [MBit/s]	2

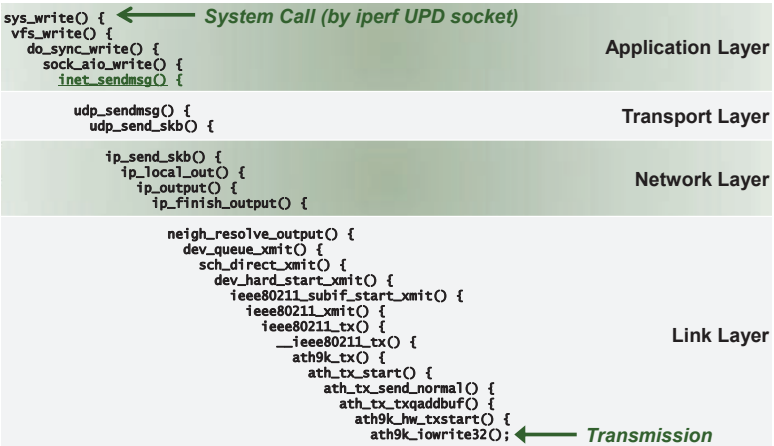

Figure 6.5: Linux kernel traversal of an UDP packet.

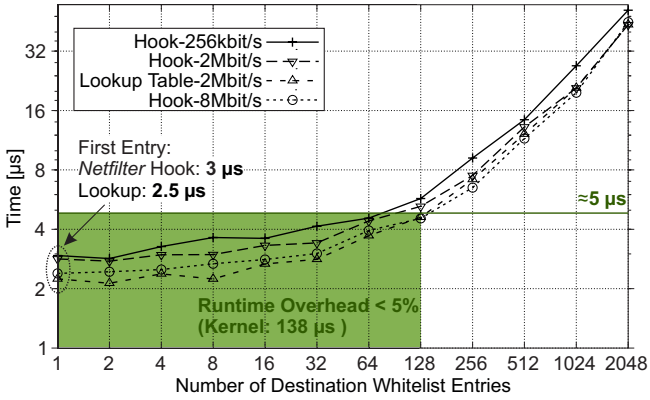

Figure 6.6: Lookup delay of the destination whitelist table [Sbe13a].

6.2.3.3 Scenario II: Destination Whitelist Lookup Delay

This scenario is similar to Scenario I, except that ROUTE-O-MATIC was now loaded into the kernel of MESHUNIT01. An entry for MESHUNIT02 is added to the Destination Whitelist of ROUTE-O-MATIC (i.e., all packets reach their destination in both directions, like in Scenario I). With ROUTE-O-MATIC running, this scenario represents a state in which a valid route for a destination exists, thus, only the Packet Monitor of ROUTE-O-MATIC is mainly involved, and the goal is to investigate its delay (i.e., the delay of the Hook function including the Destination Whitelist lookup function). Here, the entry for MESHUNIT02 was located last in the Destination Whitelist in order to evaluate the worst case (which is an iteration over all table entries), variable Destination Whitelist sizes were considered, and different UDP traffic loads were sent from MESHUNIT01 to MESHUNIT02, as illustrated in Table 6.2.

Results: The results of this experiment are depicted in Figure 6.6. The figure shows that the delay for looking up the first entry in the Destination Whitelist is only around 2.5 μs (in case of a traffic load of 2 Mbit/s). As the figure shows, this lookup time represents the main delay of the whole ROUTE-O-MATIC processing time in this scenario (i.e., the delay of *Netfilter* hook), which is around 3 μs. For up to 128 table entries, the total delay remains below 5.5 μs. This is an overhead of less than 5 % compared to the kernel's processing time determined in Scenario I (138.89 μs).
The lookup runtime of Destination Whitelist sizes larger than 128 clearly shows an expected behavior: the delay grows linearly with increasing table size, since the table lookup implementation has a linear complexity $O(n)$, where n is the

number of table entries. Those large table sizes, however, are perceived to have no relevance in real-world UAV-WMN.
Figure 6.6 also depicts that the influence of the traffic load on ROUTE-O-MATIC is to a large extent negligible. Noteworthy is the decreasing delay by higher data rates. This behavior is explained by the CPU's caching mechanisms. The higher the data rate is, the higher is the probability that the instructions used for the packet processing are located in the CPU cache. This accelerates the time of fetching these instructions and thereby slightly reduces their runtime.
To sum up, the packet processing overhead of ROUTE-O-MATIC in this scenario (*Netfilter* hook) mainly depends on the table lookup time. In real-world scenarios (e.g., UAV-WMN in emergency operations), where the Destination Whitelist size is usually lower than 128, the delay of ROUTE-O-MATIC is less than 5 % compared to the kernel's total processing time of a UDP packet. Thus, the requirement of minimal time overhead is evidently fulfilled.

6.2.3.4 Scenario III: Queue Delay

In this scenario, ROUTE-O-MATIC was investigated in case there is no route for a target destination. Four different processing stages were analyzed:

1. In the first stage, ROUTE-O-MATIC received the first packet for a destination, which was not listed in its Destination Whitelist. Hence, a RREQ message was sent by the Queue Manager to *librom*, and a queue for that particular destination was created. Finally, the packet was added to this new queue.

2. In the second stage, more packets for the unkown destination were received by ROUTE-O-MATIC, thus, the Queue Manager enqueued those packets.

3. In the third stage, the queue was full, thereby, the Queue Manager dropped new incoming packets.

4. In the fourth and last stage, ROUTE-O-MATIC received a RTADD message from *librom*. This triggered the re-injection of all queued packets.

Results of the first three stages: In the first three stages, UDP packets destined to MESHUNIT02 were generated on MESHUNIT01. For each non-routable packet, the *Netfilter* hook invokes the Queue Manager by calling its main function `queue_handler`. The latter calls both functions, `check_and_send_rreq` and `enqueue_packet`. The runtime of these operations is illustrated in Figure 6.7. The ROUTE-O-MATIC processing time of the first packet with an unknown destination (the total *Netfilter* hook function runtime) is around 120 μs. This runtime is tolerable since it represents the time of sending out a RREQ (*Netlink* route request message) and creating the queue, which occurs only once per destination. More relevant are the costs of the next stage. These reflect the delay of processing the remaining

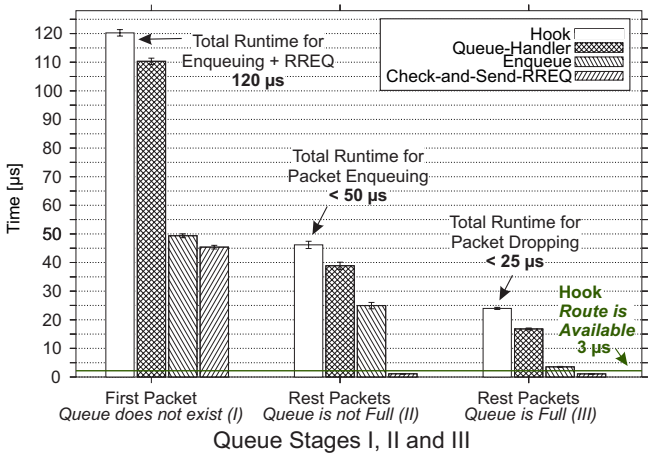

Figure 6.7: Queue delay [Sbe13a].

packets for the unknown destination. This delay is critical the higher the data rates are, since ROUTE-O-MATIC must keep up with those rates. As Figure 6.7 shows, in the second stage, the overall ROUTE-O-MATIC delay is less than 50 μs, allowing theoretically to process 20,000 packets per second. This corresponds to 235.2 Mbit/s by a packet size of 1470 Byte. The results of the second stage also reflect the decrease of the time cost of `check_and_send_rreq`, since in this stage a RREQ was already triggered. Also the runtime of `enqueue_packet` decreases around 50 %, because the queue was also already created. Once the queue was full ROUTE-O-MATIC entered the third stage. Here, the processing time decreases even more. The whole process of checking if the queue is full and then dropping the packet (done by `enqueue_packet`) needs less than 5 μs.
Once again Figure 6.7 sheds the light on the efficiency of ROUTE-O-MATIC in case a route exists (3 μs), which is highlighted by the red dashed line.

Results of the fourth stage: To determine the time needed to re-inject all packets of a full queue in the fourth stage, an extra series of measurements were performed. The queue size was set to different values ranging from 1 to 4096 packets. Figure 6.8 shows the total amount of time needed to release all packets of a full queue upon the receipt of a RTADD message (i.e., the costs of `route_add` including the release of the packets from the queue, `release_queue_for_destination`). For a default queue size of 512 packets, the whole queue is released in less than 10 ms. This corresponds to a release time of 20 μs per packet. This cost per packet slightly decreases by higher number of queued packets. It gets less than 10 μs by releasing 4096 packets. Bearing in mind that an average RTT is 1.61 ms, these results emphasize the efficiency of ROUTE-O-MATIC.

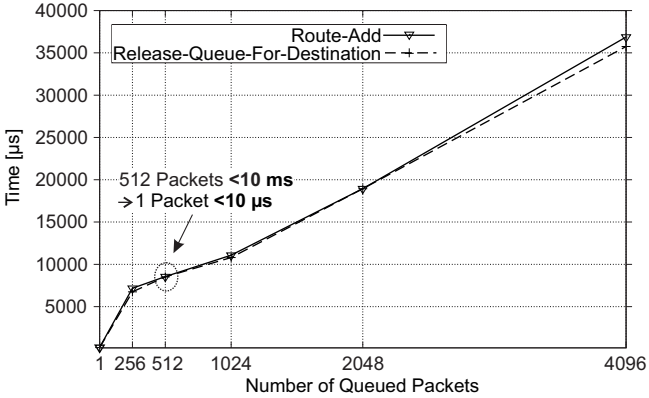

Figure 6.8: Release runtime of a full queue [Sbe13a].

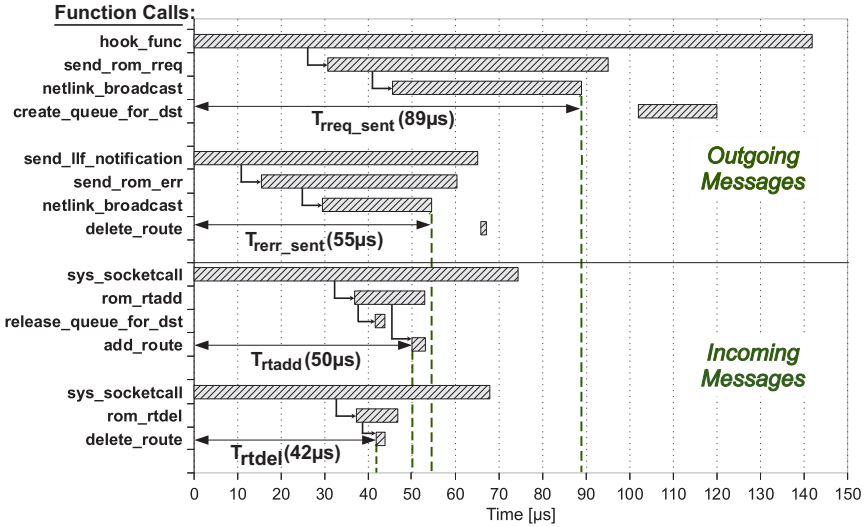

Figure 6.9: *Netlink* delay [Sbe13a].

6.2.3.5 Scenario IV: *Netlink* Delay

In this scenario, the delay of the *Netlink* communication between the ROUTE-O-MATIC kernel module and its user space library *librom* was analyzed — There are outgoing (*RREQ* and *RERR*) and incoming messages (*RTADD* and *RTDEL*).

Results: Figure 6.9 depicts the function call graphs of those messages as well as their delay.

- **Outgoing** *Netlink* **Messages:** T_{rreq_sent} and T_{rerr_sent} represent the average times for sending out a RREQ and RERR *Netlink* messages, respectively. In case of RREQ (T_{rreq_sent}), it takes ROUTE-O-MATIC around 90 μs from getting invoked by the *Netfilter* hook until the transmission function `netlink_broadcast` finishes. A RERR message indicating a transmission failure detection is even faster. In nearly half the time, after a delay of around 55 μs (T_{rerr_sent}), a RERR notification is sent out by ROUTE-O-MATIC to the *Netlink* bus.
- **Incoming** *Netlink* **Messages:** T_{rtadd} and T_{rtdel} denote the average delays for the invocation of the functions `add_route` and `delete_route`, after which, the requested action is performed. In both cases, each *Netlink* message takes less than 40 μs from entering the kernel space through `sys_socketcall` until the corresponding ROUTE-O-MATIC *Netlink* packet handler (`rom_rtadd` or `rom_rtdel`) is invoked. Afterwards, the *Netlink* message payload is interpreted and the corresponding command is triggered. In case of a RTDEL message, the `route_delete` function is called. In case of RTADD, first the queue is released and then `add_route` is called. As a result, T_{rtadd} (50 μs) is slightly higher than T_{rtdel} (42 μs).

Last but not least, all the ROUTE-O-MATIC services but the LLF were also successfully tested on the Gumstix Overo Fire COM (ARM Cortex-A8 CPU: OMAP 3503 running at 600 MHz with 512 MByte DRAM). Here, the runtime of the incoming and outgoing messages but the RERR, which depends on the LLF, were analyzed. The results showed proportional but slightly higher costs of those messages: $T_{rreq\_sent} = 122.92$ μs, $T_{rtadd} = 61.01$ μs, and $T_{rtdel} = 48.24$ μs. This is justified by the lower frequency of Gumstix-ARM.
Those tests have manifested that ROUTE-O-MATIC is to a large extent architecture-independent and, once again, it has been attested that the delay of this framework is very low.

6.2.3.6 Summary

ROUTE-O-MATIC consists of six different components: Packet Monitor, Destination Whitelist, Queue Manager, Netlink Server and Link Layer Feedback form the ROUTE-O-MATIC kernel module, while ROUTE-O-MATIC user space library *librom* provides a simple API for user space routing logics. The ROUTE-O-MATIC implementation was evaluated on the reference mesh unit Roboard RB110 to appraise its time overhead. The results show that ROUTE-O-MATIC adds an overhead of less than 5.5 μs to the processing time of each routable packet in case

of relevant Destination Whitelist table sizes (< 128 entries). Compared to the normal kernel processing time, this is only an increase of less than 5 %. Queuing a non-routable packet takes less than 50 μs on average, once a queue is created for a destination. A RREQ notification to the user space has an average delay of around 90 μs until it is sent out. The Link Layer Feedback-triggered RERR notifications take only 60 μs on average. This is by far lower than using hello messages to detect broken routes, which impose in most cases a delay higher than 1 s. Commands from the routing logic for adding and deleting a destination can be processed within 50 μs. Putting all these measurements in the context of common network delays, for instance typical packet round-trip times (1.61 ms on average), the overhead of ROUTE-O-MATIC is negligible.

6.2.4 Validation of the Feasibility of PASER

The goal of the PASER implementation in practice is mainly twofold:

1. Proving the feasibility of PASER and providing a reference implementation, which is a major missing aspect of the secure routing proposals in the literature, as none of them has a running implementation in practice;
2. Demonstrating the benefits of PASER in comparison to the IEEE 802.11 i/s security frameworks.

The PASER experimental code is open source, and it is available on the official website *www.paser.info*. The feasibility of the protocol and its benefits have been presented in different events and international conferences. These include, but are not limited to, the Vodafone innovation days [Sbe14a], this author's research work on the security and routing in wireless mesh networks [Sbe14b], including the experimental security analysis of the wormhole and blackhole attacks in Section 7.2, and the early demonstrator of the SecInCoRe research project [Kuh15, SEC]. Next, the first occasion is covered in more detail.

Secure Routing and Dynamic Key Distribution using PASER at the Vodafone Innovation Days. PASER was successfully demonstrated at the Vodafone innovation days on 4^{th}, 5^{th}, and 13^{th} November 2014 at the Vodafone Campus in Düsseldorf, Germany. Two showcases were presented using the network setup illustrated in Figure 6.10.

Showcase 1: In this showcase, a process-oriented, ad-hoc, secure, and reliable emergency wireless mesh network was deployed using the intelligent hose couplings presented in [Wol12] and the PASER protocol. A rescue fighter, wearing WLAN-capable camera glasses (see Figure 6.10, left), transmitted in real time a video from the incident scene to the decision maker in the

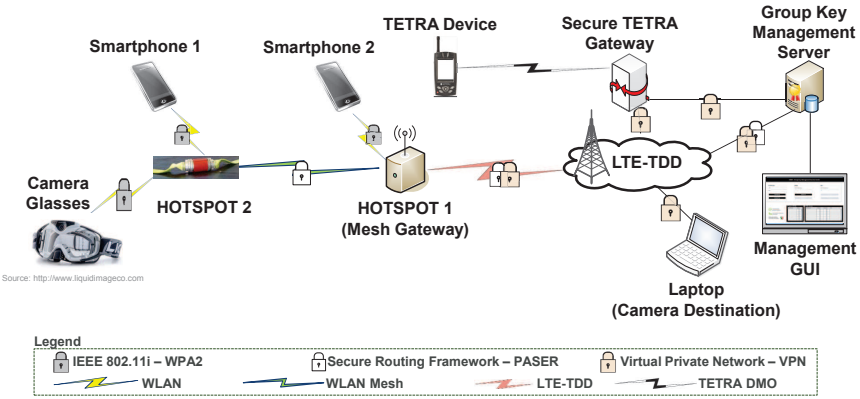

Figure 6.10: Network setup at the Vodafone innovation days.

control room (i.e., to the laptop in Figure 6.10, right). The video was sent over the PASER-based WLAN mesh network to the WLAN mesh gateway, which forwarded it over LTE-Time Division Duplex (LTE-TDD) to the decision maker. The WMN was deployed in periods of four hours, and the video streamed seamless. This emphasizes the feasibility of PASER and its possibility to run reliably for relatively long durations, e.g., in UAV-WMN.

Showcase 2: In this showcase, the dynamic refreshment of the WMN keys using PASER was demonstrated. As proposed in [Sbe11a], a QR-coded WLAN access key was pushed via the secure Terrestrial Trunked Radio (TETRA) network to the rescue teams. Volunteers brought their own device (smart phones in Figure 6.10), scanned the QR-code, and got access to the network. Using novel solutions (e.g., trusted smart phone applications and a TETRA-IP API), a secure TETRA-IP chat group was established. In this way, the volunteers could offer better support to rescue fighters for an improved management of the crisis.

At the network side, the PASER protocol was used to distribute the same key to the mesh access point (i.e., the CTK in Chapter 5). If a volunteer should be excluded from the network, or the key should be periodically refreshed due to security reasons, it was demonstrated how a new QR-coded key was pushed via TETRA to the rescue teams, in the meantime, the key was distributed on-the-fly via PASER to all WLAN mesh access points. The successful demonstration of this showcase proves that PASER has solved the interdependency cycle between secure routing and symmetric key distribution, which is a major asset in comparison to related works.

7

Security Analysis of PASER

Contents of this Chapter

In this chapter, the impact of the blackhole and wormhole attacks on the performance of WMNs in practice is analyzed. Afterwards, a security comparison of PASER and the three alternate solutions, HWMPS, BATMANS, and ARAN, is given. The results show that PASER is more resilient to routing attacks than its counterparts in UAV-WMN.

Publications. Parts of this chapter have been presented in [Sbe14b] and [Sbe15].

7.1 Experimental Analysis of the Blackhole and Wormhole Attacks

In this section, an experimental testbed of the blackhole and wormhole attacks is set up, and the robustness of PASER against both attacks in comparison to the non-secure routing protocols HWMP and BATMAN combined with the security frameworks of IEEE 802.11s/i (i.e., HWMPS and BATMANS) is analyzed. It is investigated whether these solutions are suitable to be deployed in security-critical scenarios, such as UAV-WMN.

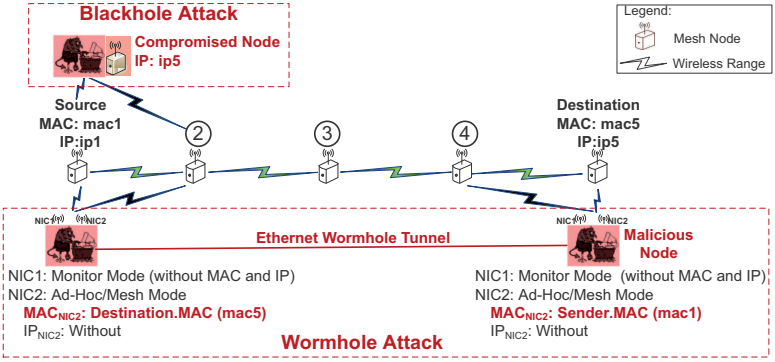

Figure 7.1: Experimental testbed of the wormhole and blackhole attacks [Sbe14b].

7.1.1 Experimental Blackhole Attack

The blackhole attack was implemented in an experimental testbed consisting of a static chain of five mesh nodes and an attacker node, as depicted in Figure 7.1 (top). In a blackhole attack, assuming the attacker has compromised a node or a key, the attacker impersonates the destinations (in general, the gateways in WMNs). Implementing the attack is quite straightforward as the attacker only needs to set up the same IP address as the destination, and the routing protocol will take care of the rest. In case of MAC-based protocols such as HWMPS, ARP spoofing is also required. The latter can be carried out with a simple *ping* command. Unfortunately, not only direct neighbors are affected by the blackhole attack but also other nodes located outside the transmission range of the attacker because the routing protocol will disseminate the fake identity of the attacker throughout the whole network. As a result, the attacker can redirect the traffic and sabotage the network. In UAV-WMN, the attacker might also propagate fake information to hijack the UAVs.
In this experiment, the Roboard RB110 mesh node was used, on which HWMPS and BATMANS were installed (see Section 4.3.2) and PASER was implemented. The network was operated according to IEEE 802.11n using a 40 MHz frequency bandwidth. The source, mesh node 1, sent a 3.5 Mbit/s constant bit rate UDP traffic to the destination, mesh node 5. The attacker impersonated mesh node 5 and aimed to sabotage the network. Five repetitions were run for each protocol. Selected results are depicted in Figure 7.2. As the figure shows, in less than one minute after starting the attack, the non-secure routing protocols combined with the IEEE 802.11 security frameworks, HWMPS and BATMANS, were get caught in the attacker's trap. That is, the IEEE 802.11 security frameworks are not able

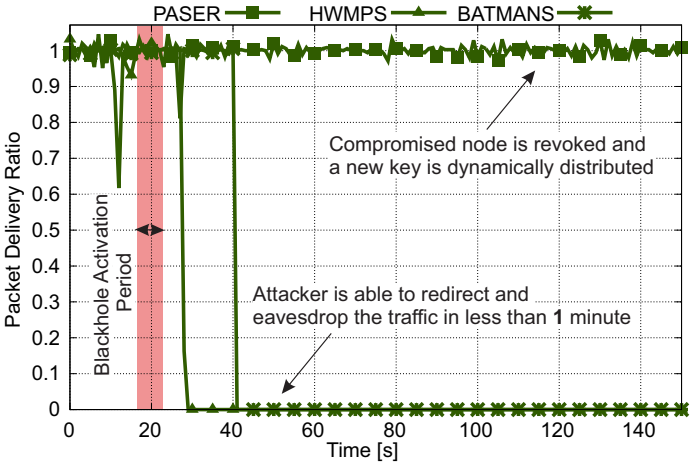

Figure 7.2: Impact of the blackhole attack on the PDR in PASER, HWMPS, and BATMANS in the experimental testbed [Sbe14b].

to mitigate the attack. In case the attack is carried out, the only way the network operator can react on it is to collect all nodes, reconfigure them, and to set up the network again. In contrast, in case of the secure routing protocol PASER, due to its dynamic key management scheme and its security features, the compromised node was revoked, and the network keys were dynamically refreshed. Thereby, the performance of the network is not affected by this attack in case of PASER, as depicted in Figure 7.2.

7.1.2 Experimental Wormhole Attack

To carry out the wormhole attack, two malicious nodes directly connected with each other (e.g., using LAN or directional antennas) are placed at the incident scene, as depicted in Figure 7.1 (bottom). These nodes transparently forward routing messages between distant nodes (i.e., from one area of the network to another) faster than the legitimate nodes. This causes affected nodes located in the different areas to believe they are neighbors. Consequently, they start sending messages to each other via the wormhole tunnel instead of using the legitimate relay nodes. As a result, the attacker can drop all the data packets causing a sabotage of the network.
In contrast to the blackhole attack, the attacker in a wormhole attack does not require network access. To implement the attack, the attacker just needs two

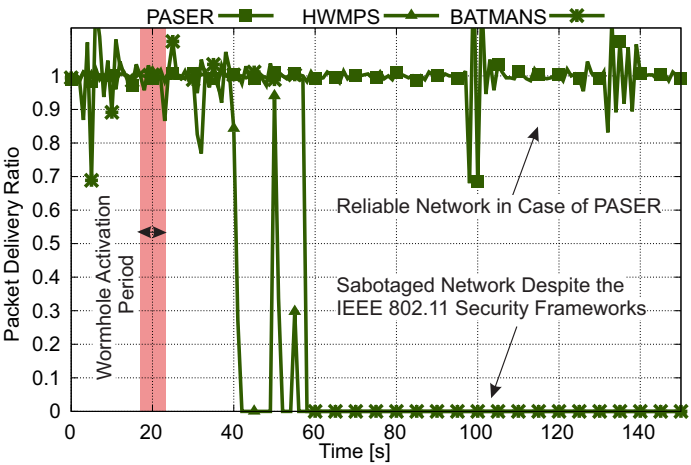

Figure 7.3: Impact of the wormhole attack on the PDR in PASER, HWMPS, and BATMANS in the experimental testbed [Sbe14b].

interfaces per node: one is set in monitor mode to eavesdrop all the frames in the proximity regardless of their destination, and one is set in ad-hoc or mesh mode to automatically reply with an ACK frame to any unicast data frame sent by the victim nodes. In this way, the victim nodes never detect that the network is sabotaged. For instance, in the network in Figure 7.1 (bottom), when mesh node 1 sends a data packet to mesh node 5, the attacker's node in the proximity of mesh node 1 drop the corresponding frame, yet it acknowledges its successful receipt by sending back an acknowledge frame on behalf of mesh node 5. To sabotage the network, the attacker only forwards routing messages to manipulate the routing topology, and the attacker discards data packets. In case the IEEE security frameworks are running, routing messages and UDP/TCP data packets are both encrypted in data frames, thus, the attacker cannot decide based on the content whether to forward the frame or not. However, as the size of routing messages is typically < 500 Byte, the attacker can decide based on the size of the frame whether to forward it or not (advanced attackers can also take the frame's frequency into consideration).
In this experiment, see Figure 7.1 (bottom), the attack was run on a static chain of five nodes using the Roboard RB110, on which HWMPS and BATMANS were installed (see Sub-section 4.3.2) and PASER was implemented. The network was operated according to IEEE 802.11n and a 40 MHz frequency bandwidth. The source, mesh node 1, sent a 3.5 Mbit/s constant bit rate UDP traffic to the destination, mesh node 5. One attacker's node was placed in the proximity of

mesh node 1, the second attacker's node was placed in the proximity of mesh node 5, and both nodes were configured appropriately to run the wormhole attack. Five repetitions were run for each protocol. Selected results are depicted in Figure 7.3. As the figure shows, non-secure routing protocols in combination with the IEEE 802.11 security frameworks are prone to the wormhole attack. How fast the wormhole is established depends on the routing protocol design and metric. In case of BATMANS, the route is selected based on the number of OGMs received within a sliding interval, thus, it takes a bit longer time than HWMPS to use the wormhole tunnel. In the latter case, the success of the attack is very fast, as HWMPS in practice sends a route request to refresh active routes every four seconds. Only PASER is robust against the wormhole attack, due its geographical leash mechanism and its security features.

7.2 Security Comparison

In this section, a security comparison between PASER and the following three representative alternate solutions is given:

1. HWMPS: A combination of the security mechanisms of the IEEE 802.11s mesh standard and the HWMP routing protocol, which is specified in the mesh standard.
2. BATMANS: A combination of the IEEE 802.11i security mechanisms and the BATMAN routing protocol, which is widely deployed in community networks [BATc].
3. ARAN: The well-known, reactive, and secure routing protocol ARAN, which is considered to date (see, e.g., [Pie14, Mah13]) as an exemplary solution for secure routing in wireless multi-hop networks.

Table 7.1 illustrates a security comparison between PASER and its alternatives. It provides a mapping between each attack of the attacker model presented in Sub-section 5.1.2, the security goals that must be achieved to combat the attack, and whether the considered solutions guarantee these goals. Table 7.2 gives an overview of the mechanisms implemented by each solution to fulfill the security goals. The information provided in both tables are based on the experimental analysis in the previous section and on the works in [Sen13, Sgo13, Abu08].
As Table 7.1 shows, in case of an external attacker, all solutions protect against internal and time-base replay attacks, as all solutions achieve message authentication and freshness. Only PASER, however, is able to fulfill neighbor authentication. Thus, only PASER protects against the position-based replay and wormhole attacks. This fact is verified in the experimental analysis in Sub-section 7.1.2. In case of an internal attacker, the IEEE 802.11s/i standards do not provide any

Table 7.1: Comparison of mitigated attacks [Sbe15].

+: Attack is disabled (proactively), Δ: Attack is prevented (reactively), -: Attack is possible, o: It depends

Attacker Type	**Attack Name**	**Security Goals**	**PASER**	**ARAN**	**HWMPS, BATMANS**
External	Internal attacks	Message authentication	+	+	+
	Time-based replay	Message freshness	+	+	+
	Position-based replay	Neighbor authentication	+	-	-
	MAC impersonation	MAC address authentication	-	-	-
	Wormhole	Neighbor authentication	+	-	-
		MAC address authentication			
Internal	Flooding & path diversion	Intrusion detection & dynamic key management	Δ	o	-
	IP impersonation	IP address authentication	o	o	-
	Fabrication	Origin authentication	+	+	-
		Intrusion detection & dynamic key management			
	Blackhole	IP address authentication	o	o	-
		Intrusion detection & dynamic key management	Δ	o	

Table 7.2: Comparison of security mechanisms [Sbe15].

Security Goals	**PASER**	**ARAN**	**IEEE 802.11s/i**
Message authentication	Digital signature of sender (untrusted neighbors)	Digital signature of sender	CCMP
	HMAC (trusted neighbors)		
Message freshness	Nonce (untrusted neighbors)	Nonce	CCMP
	Nonce or one-time authentication secrets (trusted neighbors)		
Neighbor authentication	Digital signature and position information of sender (untrusted neighbors)	-	-
	One-time authentication secrets and position information of sender (trusted neighbors)		
Origin authentication	Digital signature of originator	Digital signature of originator	-
Dynamic key management	Key management scheme	Broadcast of certificate's revocation	-

CCMP: Counter mode Cipher block chaining Message authentication code Protocol

protection, as the sole security credential used in their security frameworks (personal mode) is the pre-shared key. Once this key is compromised, the attacker can successfully mount all internal attacks. In case of ARAN, digital signatures are used to combat the fabrication attack, see Table 7.2. If the IP address is bound to the public key, the digital signature also protects against the impersonation and blackhole attacks. To further mitigate internal routing attacks, ARAN implements a broadcasting mechanism of revoked certificates to exclude compromised nodes. ARAN, however, does not address the case of transmission errors of these revocation broadcast messages due to interference or channel errors.

In comparison to ARAN, PASER mitigates the same internal attacks and offers a more fail-safe *dynamic key management* scheme. PASER guarantees the detection of nodes or key revocation despite non-successful reception of a revocation broadcast message. As long as any neighbor or any node along a requested route between source and destination is aware of the revocation, a PASER node detects it (see Section 5.3.5). In UAV-WMN, this is always the case as at least the gateway at the ground station is aware of any revocation process, and this gateway is a common destination for all the UAVs. In comparison to the IEEE 802.11s/i, PASER's symmetric scheme offers two levels of security: even if the attacker compromises the symmetric group transient key, the attacker will not be able to impersonate a node or mount a blackhole attack as the attacker cannot generate one-time authentication secrets on behalf of legitimated nodes.

8

Performance Analysis of PASER

Contents of this Chapter

This chapter deals with the performance evaluation of PASER and three alternatives in realistic UAV-WMN scenarios. The following steps are taken:

- A theoretical and simulation-based analysis of the route discovery delay of PASER and its alternatives is performed. This delay constitutes, along with the message overhead of the protocols, for which asymptotic expressions are provided, the main impact on the overall network performance.
- The performance of the protocols is evaluated in UAV-WMN-assisted network provisioning and area exploration scenarios, among others. Here, OMNeT++ and INETMANET, realistic mobility patterns of UAVs, and an experimentally derived channel model of the air-to-air link between the UAVs are used. Both the route discovery delay and the message overhead of the protocols are taken into consideration to justify the results.

Publications. Parts of this chapter have been presented in [Sbe15].

8.1 Analysis of the Route Discovery Delay

The route discovery delay is the time needed to find a route to an unknown destination. It is composed of the time needed to transmit the required routing messages, $Cost_{Comm}$, and the time needed to process these messages, $Cost_{Comp}$. The definition of this delay is given in Equation 8.1, which is based on the notations in Table 8.1.

$$Delay_{RD}(D, I^*)_{[s]} = Cost_{Comm(D,I^*)_{[s]}} + Cost_{Comp(D)_{[s]}} \tag{8.1}$$

Definition 2. *Let the route discovery be the case where the sender does not have or lost the routes to the next hop and the corresponding destination, and it needs to (re)discover the latter provided that intermediate nodes have the route.*

8.1.1 Lower Bound for the Communication Costs

A lower bound equation of $Cost_{Comm}$ to find that route is depicted in Table 8.2. In this equation, it is assumed that the transmission of the routing messages is always successful as the main goal is to compare the efficiency of the security schemes of the protocols when the route length increases, and whether thereby certain latency requirements are violated. Non-successful transmission of the routing messages is dependent on the network topology, traffic load, mobility, and channel characteristics, and it is less related to the security scheme used. It leads to link layer retransmissions and route timeouts, which is investigated in simulation.

As Table 8.2 shows, in case of the reactive protocols PASER and ARAN, in contrast to HWMPS, intermediate nodes cannot reply on behalf of the destination due to security reasons. In this respect, PASER mainly relies on unicast messages to contact the destination, while ARAN uses broadcast messages. In case of the proactive protocol BATMANS, at least two periodic messages (called *OGMs*) must be exchanged with the next hop to consider the link as valid. Only then, *OGMs* of the destination received through that next hop are processed.

The transmission time of the unicast messages in the equations of Table 8.2 can be calculated according to Sub-section 4.3.1. The transmission time of the broadcast messages can be analogously calculated taking into consideration the use of basic PHY data rates (typically 1 Mbit/s in case of IEEE 802.11g), and that broadcast messages are not acknowledged. The message sizes needed to calculate the transmission time are provided in Table 8.2 (bottom). Here, a Merkle tree consisting of 2^{14} secrets, a secret size of 32 Byte, and a certificate size of 701 Byte are assumed.

Table 8.1: Notations used in the analysis of the route discovery delay [Sbe15].

Notation	Description	Notation	Description
σ	Estimated average elapsed intervals	I^*	Interval of sending periodic messages; *: only in case of proactive protocols
$Cost_{Comm}$	Communication costs	MIC	Message integrity code
$Cost_{Comp}$	Computational costs	Msg, Msg_B, Msg_U	All messages, broadcast messages, unicast messages
D	Diameter: number of links on a route	$Sign_{Gen/Ver}$	Generation / verification of digital signature
Dec/Enc	Decryption / encryption	T_U	Transmission time of unicast frames
$Delay_{RD}$	Route discovery delay	T_B	Transmission time of broadcast frames
$Guard_B$	Uniform random time (0, 0.005 s), waited before the transmission of broadcast messages	T_{Op}	Computation time of cryptographic operations
$MAC_{Gen/Ver}$	Generation / verification of keyed-hash message authentication code	#	Number of

8.1.2 Lower Bound for the Computational Costs

The equation of $Cost_{Comp}$ to find the route and the time costs of the corresponding cryptographic operations are depicted in Table 8.3. These costs (see [Sbe14b, Sbe12b] for details about the implementation of the operations) are experimentally measured using ftrace, as in [Sbe13a], on the Roboard RB110 embedded system. Table 8.3 emphasizes the efficiency of the PASER security scheme as it illustrates that in case the route length increases, efficient MAC operations (0.248 ms) are used, i.e., only these depend on D) while inexpensive signature operations (28.595 ms) are applied in ARAN.

Table 8.2: Calculation of lower bound of $Cost_{Comm}$ in Equation 8.1 [Sbe15].

$$\mathbf{Cost_{Comm}(D, I^*)_{[s]}} = \sum_{\mathbf{Node=1}}^{\mathbf{D+1}} \Big(\sum_{\mathbf{i=1}}^{\mathbf{\#Msg_B(Node)}} (\mathbf{T_B(Size_{Msg})_{[s]}} + \mathbf{Guard_{B\,[s]}}) + \sum_{\mathbf{i=1}}^{\mathbf{\#Msg_U(Node)}} \mathbf{T_U(Size_{Msg})_{[s]}} \Big) + \sigma \mathbf{I^*_{[s]}}$$

Protocol	**Costs [s]**
PASER (reactive)	$T_{B_{UB-RREQ}} + Guard_B + (D-1)T_{U_{TU-RREQ}} + (D-1)T_{U_{TU-RREP}}$
ARAN (reactive)	$T_{B_{RREQ_{Sender}}} + Guard_B + (D-1)\,(T_{B_{RREQ_{Inter.}}} + Guard_B) + T_{U_{RREP_{Destination}}} + (D-1)\,T_{U_{RREP_{Inter.}}}$
HWMPS (reactive part)	$T_{B_{PREQ}} + Guard_B + T_{U_{PREP}}$
BATMANS (proactive)	$1.75\,I + (4+D)\,T_{B_{OGM}} + (4+D)\,Guard_B$

Message sizes of the protocols - Required to calculate T_B and T_U in $Cost_{Comm}$

Message	**Size [Byte]**	**Message**	**Size [Byte]**
PASER			
UB-RREQ	1066	UU-RREP	1090
TU-RREQ	714	TU-RREP	709
TU-RREP-ACK	538		
ARAN			
$\text{RREQ}_{\text{Sender}}$	860	$\text{RREQ}_{\text{Intermediate}}$	1693
$\text{RREP}_{\text{Destination}}$	852	$\text{RREP}_{\text{Intermediate}}$	1685
HWMPS			
PREQ: Path request	27	PREP: Path reply	27
BATMANS			
OGM: Originator message			16

Table 8.3: Calculation of lower bound of $Cost_{Comp_{[s]}}$ in Equation 8.1 [Sbe15].

$$\mathbf{Cost_{Comp}(D)_{[s]}} = \sum_{\mathbf{Node=1}}^{\mathbf{D+1}} \sum_{\mathbf{Msg=1}}^{\mathbf{\#Msg(Node)}} \sum_{\mathbf{Op=1}}^{\mathbf{\#Op(Msg)}} \mathbf{T_{Op_{[s]}}}$$

Protocol	**Costs [s]**
PASER (reactive)	$Nonce_{Gen} + 2Sign_{Gen} + Sign_{Verf} + (D-1)(MAC_{Verf} + MAC_{Gen}) + Sign_{Verf} + Sign_{Gen} + MAC_{Gen} + (D-1)(MAC_{Verf} + MAC_{Gen}) + MAC_{Verf} + Sign_{Verf}$
ARAN (reactive)	$Nonce_{Gen} + Sign_{Gen} + (D-1)Sign_{Gen} + (2D-3)Sign_{Verf} + 2Sign_{Verf} + Sign_{Gen} + (D-1)Sign_{Gen} + (2D-3)Sign_{Verf} + 2Sign_{Verf}$
HWMPS (reactive part)	$2(Enc + MIC_{Add} + Dec + MIC_{Ver})$
BATMANS (proactive)	$(4 + D)(Enc + MIC_{Add} + Dec + MIC_{Ver})$

Average time costs of the cryptographic operations in $Cost_{Comp}$ - Measured on the Roboard RB110 [ROB] (30 runs)

Cryptographic operation	**Time [ms]**	PASER	ARAN	HWMPS	BATMANS
Signature generation	27.021	x	x		
Signature verification	1.574	x	x		
Nonce generation	0.432	x	x		
MAC generation/verification	0.141	x			
Encryption (hw)	0.089			x	x
Decryption (hw)	0.072			x	x
MIC generation/verification (hw)	0.002			x	x

hw: hardware accelerator

8.1.3 Evaluation of the Route Discovery Delay

Using Table 8.2 and Table 8.3, the route discovery delay of the protocols can be calculated according to Equation 8.1, and it can be determined whether their security schemes are a limitation with respect to latency in large networks. This delay is calculated for route lengths up to 19 links. The IEEE 802.11g technology is considered. The PHY data rate is set to 11 Mbit/s. The basic PHY data rate is 1 Mbit/s. The UDP and IP header sizes are 8 and 20 Byte, respectively. The MAC header size is 34 Byte; in case of HWMPS, it is 38 Byte. The MAC header size is increased by 16 Byte when the IEEE 802.11s/i security mechanisms are used. The BATMANS's OGM-interval is set to 0.5 s, according to the findings in [Sbe14c]. The route discovery delay is also evaluated in OMNeT++ [Var08] and its INETMANET framework. The goal of this simulation-based evaluation is twofold: 1) to validate the derived equations in case of successful transmission of routing messages, and 2) to investigate the impact of non-successful transmissions on the route discovery delay. Here, 10 % and 20 % unsuccessful transmission rates, i.e., Frame Error Rate (FER), are considered, respectively.
Figure 8.1 depicts the results of this analysis. As the figure shows, the equations' results match the performance of the protocols in simulation. This attests the validity of these equations. The results show only a slight increase of the route discovery delay of PASER while the route length is multiplied, and the FER is increased. This sheds light on two facts: First, the security scheme of PASER is not a limitation with respect to latency in case of long routes, in contrast to that of ARAN, see Figure 8.1 (top right) in case of 0 % FER. A comparison of the delay of both protocols given a route length of 19 links highlights the efficiency of the security scheme of PASER, which is ten times faster than ARAN in that case. Second, the route discovery mechanism of PASER, relying on unicast messages, is robust against high FERs. In contrast, that of ARAN, which relies on broadcast messages to wider propagate the route discovery information, is ineffective in case of high FER. If the messages are not successfully received, they will not be retransmitted, thereby, route timeouts occur, after which a new route request must be started. Figure 8.1 (bottom, left) emphasizes the lightweight of HWMPS —At the expense of the security level it can achieve, see Table 7.1 in Section 7.2. Figure 8.1 (bottom, right) indicates that the delay in BATMANS is mainly caused by the periodic message interval as σ intervals ($\sigma \in \mathbb{R}\ and\ \sigma > 1$) must be waited in order to have exchanged the necessary messages, and in case of high FER, these message can get lost, leading to a considerable increase in the delay. On a final note, while the results in Figure 8.1 are generated using IEEE 802.11g, the delay of PASER decreases to more than 20 % when using IEEE 802.11n/ac, given route lengths higher than 5 links. That is, by using these recent technologies, PASER is able to meet the multimedia latency requirement of 150 ms [Abo03] even in case of route lengths consisting of 19 links and a FER of 20 %.

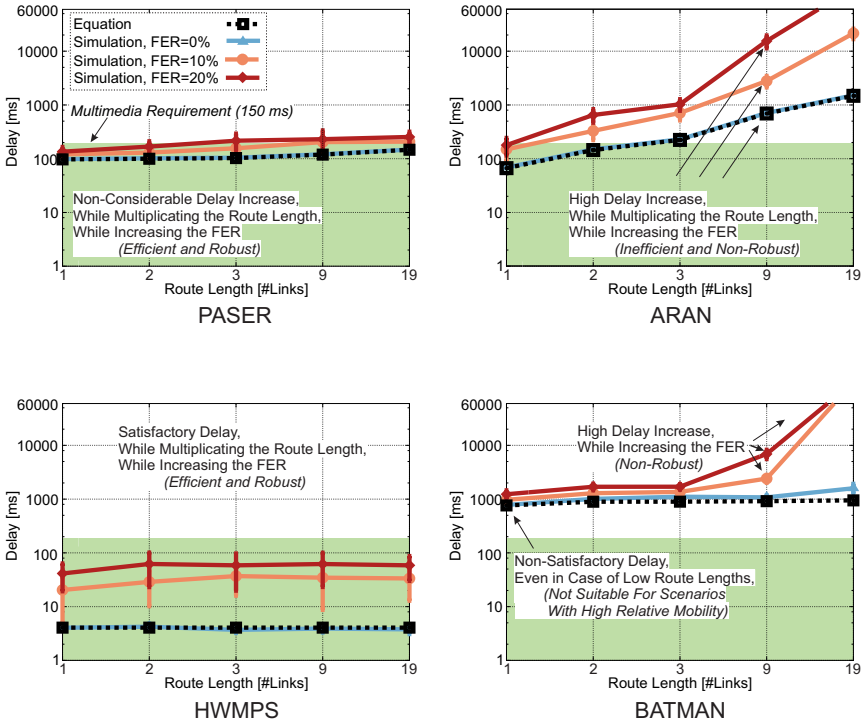

Figure 8.1: Time for finding a route to a given destination ($Delay_{RD}$) in theory and in simulation [Sbe15].

8.2 Asymptotic Message Overhead

In this section, the message overhead of the routing protocols is estimated as a function of limiting parameters with respect to the network performance, based on the work in [Vie04]. These parameters are the network size N, the average route length L, the average number of one-hop neighbors (node density) Δ, the average number of active routes (traffic flows) per node α, and the mobility (link breakage rate) μ. While the work in [Vie04] only focused on the overhead of broadcast messages, an approximation of the overhead of unicast messages is also provided in this research as PASER mainly relies on these messages. Let h_r be the *hello* rate (i.e., that of PASER) and t_p the topology broadcast rate (i.e., *OGM* rate in case of BATMANS and proactive *PREQ* rate of the root

Table 8.4: Message overhead of the routing protocols [Sbe15].

Protocol	Broadcast messages rate	Unicast messages rate
PASER	$\mu \cdot \alpha \cdot L \cdot N + h_r \cdot N$	$\Theta(\Delta \cdot \mu \cdot \alpha \cdot L^2 \cdot N)$
ARAN	$\mu \cdot \alpha \cdot L \cdot N \cdot (N-1)$	$\Theta(\mu \cdot \alpha \cdot L^2 \cdot N)$
HWMPS	$\mu \cdot \alpha \cdot L \cdot N + t_p \cdot N$	$\Theta(\Delta \cdot \mu \cdot \alpha \cdot L \cdot N)$
BATMANS	$t_p \cdot N^2$	-

element in case of HWMPS), the message overhead of the routing protocols is illustrated in Table 8.4. The table depicts the number of messages per second or an asymptotic tight bound of this overhead. This information can be used to support the analysis of the relative performance of the protocols in a certain scenario, as done in Section 8.3. It can be also used to determine the scalability of the protocols, using the method proposed in [San02]. For instance, let Ohd be the overhead of the protocols and Tr the minimum amount of bandwidth required to forward the traffic load in the network in case the routes were statically set. Let Ψ_{λ_i} be the network scalability factor with respect to a parameter λ_i and $\rho_{\lambda_i}^{Prot}$ the routing protocol scalability factor, these terms are defined in Equation 8.2 [San02].

$$\rho_{\lambda_i}^{Prot} = \lim_{\lambda_i \to \infty} \frac{\log Ohd^{Prot}(\lambda_1, \lambda_2, ...)}{\log \lambda_i}, \quad \Psi_{\lambda_i} = \lim_{\lambda_i \to \infty} \frac{\log Tr(\lambda_1, \lambda_2, ...)}{\log \lambda_i} \tag{8.2}$$

A protocol is considered scalable with respect to λ_i if $\rho_{\lambda_i}^{Prot} \leq \Psi_{\lambda_i}$. Given a constant Δ in case of an increasing network size [San02, San01] and a constant L, it is $Tr(\mu, \alpha, N) = \Theta(\alpha N)$ —Increasing L without bounds would jeopardize the performance of UAV-WMN, on which strict requirements are posed due to critical real-time telemetry transmissions [Dan10]. Thus, $\Psi_{\lambda_\mu} = 0, \Psi_{\lambda_\alpha} = 1$, and $\Psi_{\lambda_N} = 1$. Besides, in that case, $Ohd^{PASER} = Ohd^{HWMPS} = \Theta(\mu\alpha N)$, $Ohd^{ARAN} = \Theta(\mu\alpha N^2)$ and $Ohd^{BATMANS} = \Theta(N^2)$. That is, only PASER and HWMPS are scalable with respect to the network size: $\rho_N^{PASER} = \rho_N^{HWMPS} = 1 = \Psi_N < \rho_N^{ARAN} = \rho_N^{BATMANS} = 2$. All the protocols are scalable with respect to the traffic load: $\rho_\alpha^{BATMANS} = 0 < \rho_\alpha^{PASER} = \rho_\alpha^{ARAN} = \rho_\alpha^{HWMPS} = 1 = \Psi_\alpha$. Only BATMANS is scalable with respect to mobility: $\rho_\mu^{BATMANS} = 0 = \Psi_\mu < \rho_\mu^{PASER} = \rho_\mu^{ARAN} = \rho_\mu^{HWMPS} = 1$. Nevertheless, as the relative mobility of low-altitude cooperative UAVs is limited due to their small size and their communication aware mobility strategies (see Paragraph 8.3.4), the mobility factor is not a major concern in practice. Much more relevant are the network size and the traffic load, for which it has been shown that PASER is scalable.

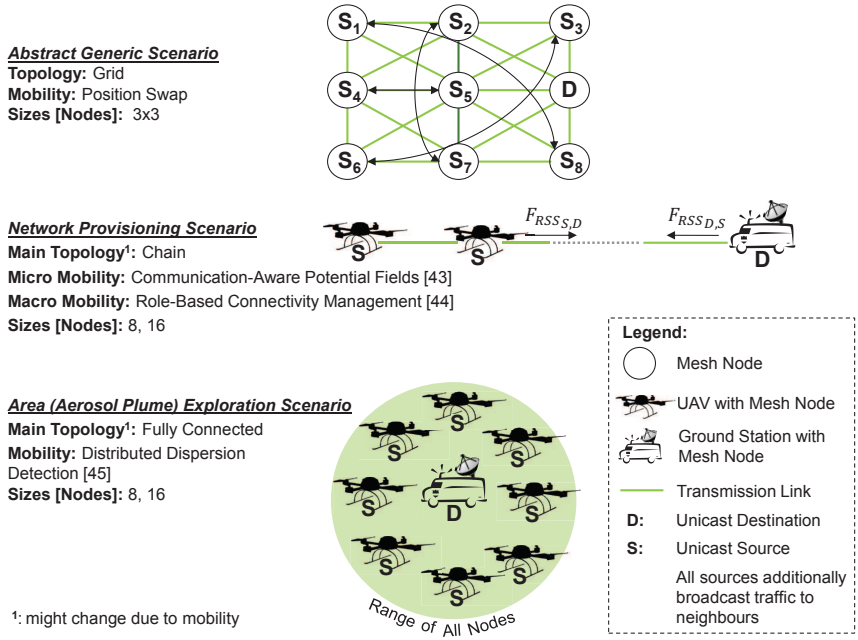

Figure 8.2: Overview of the analyzed network topologies, traffic flows, and mobility patterns [Sbe15].

8.3 Performance Evaluation

In this section, a performance evaluation of the routing protocols in OMNeT++ is presented. The corresponding topology and traffic models are first outlined. Second, the realistic UAV mobility patterns used, some of which were demonstrated in practice in [AIR, Roh10, Dan09], are elaborated. Third, an experimentally derived channel model for air-to-air UAV-WMN links is explored. This channel built the basis of the close to reality performance analysis done. Last, the results of the performance evaluation are presented.

8.3.1 Topology Models

Three network topologies are considered, as illustrated in Figure 8.2. The first is a grid topology, reflecting a generic synthetic scenario. This topology is used

to analyze the impact of link breaks (mobility) and transmission errors on the performance of the protocols. The second topology is mainly a chain of nodes, representing network provisioning scenarios. It is used to analyze the impact of the route length on the performance. The third topology is mainly a formation of fully connected UAVs, as intended to be deployed in aerosol plume exploration scenarios. It is used to analyze the impact of the node density on the performance. Different network sizes are investigated, see Figure 8.2. For each network size, the transmission range, number and duration of physical links of nodes differs. The available transmission time between any neighbor pair strongly depends on the channel model and mobility patterns used.

8.3.2 Traffic Models

Two types of traffic are considered, as depicted in Figure 8.2. First, broadcast Constant Bit Rate (CBR) traffic is periodically exchanged between neighbor-UAVs, e.g., telemetry data. Second, unicast CBR traffic is sent from all UAVs towards the gateway (the ground station). Depending on the network size, the data rates of both traffic types are adjusted so that the analyzed networks are never congested in case of HWMP (i.e., without security). Hereby, the data rates are set to the maximum value for which HWMP achieves 100 % packet delivery in the main topology of each scenario in case all nodes were static (i.e., in static grid, static chain, and static fully connected network).

8.3.3 Channel Models

Two channel models are considered. The first is a large-scale model based on the free space propagation loss. The second is a combination of the free space model and a small-scale fading model that follows a Rician distribution. The free space propagation loss can be expressed by Equation 8.3, with G_t and G_r being the antenna gains, λ the wave length, d the distance between sender and receiver, and γ_0 the attenuation coefficient, see [Fri46].

$$L_{[dB]} = 10 \log_{10} \left(\frac{1}{G_t G_r} \left(\frac{4\pi}{\lambda} \right)^2 d^{\gamma_0} \right) \tag{8.3}$$

The attenuation coefficient γ_0 typically ranges between $[2, 5]$, where $\gamma_0 = 2$ is used for free space (rural) environments, and $\gamma_0 = 5$ is used for (urban) environments with strong damping. Experimental validations, performed by Niklas Goddemeier [God], of the air-to-air UAV-WMN link using two UAVs flying at 30 m altitudes and several WLAN cards such as the DNMA-92 Atheros mini-PCI card and the TP-Link TL-WN821N mini-USB-adapter provided matching results

with the free space propagation loss for $\gamma_0 = 2.65$. Therefore, this value is used in this research. The frequency is set to 2.412 GHz, the receiver sensitivity is -91 dBm, the transmitting power is 20 dBm, and $G_t = G_r = 1$. Thus, according to Equation 8.3, a node can sense the signal in a range of 473.8 m. The Signal to Noise plus Interference Ratio (SNIR) threshold is set to 4 dB. This means, in case of -101 dBm thermal noise and 9 dB noise factor, as used in this research, the maximum transmission range can be calculated as 365.1 m.
The Rice distribution is defined according to Equation 8.4, with $x \in \mathbb{R}_+$, I_0 being the modified Basel function of zero order and the first kind, and v and s reflecting the strength of the dominant and non-dominant paths respectively, see [Dur02].

$$p_\xi(x) = \frac{x}{s^2} \exp\left(\frac{-x^2 - v^2}{2s^2}\right) I_0\left(\frac{xv}{s^2}\right) \tag{8.4}$$

Through experimental measurements, performed by Niklas Goddemeier [God], using the hardware-in-the-loop UAV testbed [God12b], approximations of the parameters v and s are determined. Figure 8.3 shows the Probability Density Function (PDF) of x in case of different transmission power levels. Each distribution represents the variation in the Received Signal Strength (RSS) measured

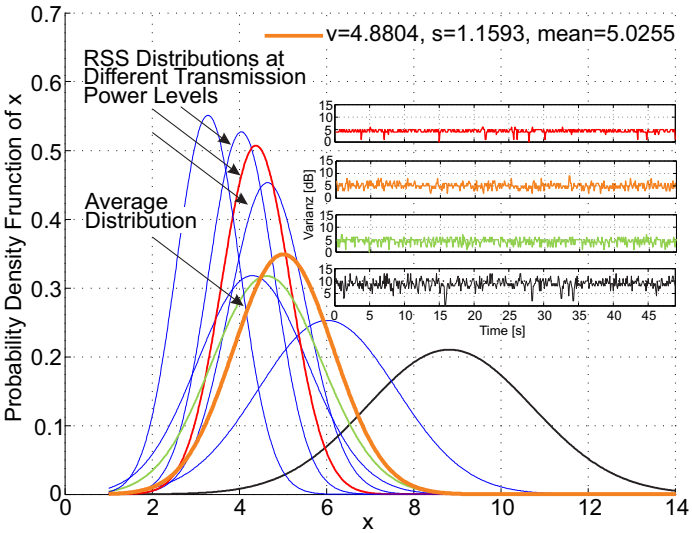

Figure 8.3: Parameters identification of a Rician channel model for UAV-WMN—derived from experiments [Sbe15].

by the communication hardware (ranging from −45 dBm to −84 dBm). Based on these results, an average distribution is derived, as depicted in Figure 8.3. Finally, to obtain a realistic channel model, large-scale and small-scale fading are combined according to Equation 8.5, with L_{total} being the total propagation loss.

$$L_{total[dB]} = L_{[dB]} - (p_\xi(x) - mean) \tag{8.5}$$

8.3.4 Mobility Patterns

In the grid scenario, apart from forced position swaps, the nodes are static. The positions of $\text{UAV}_{(N-i) \bmod N}$ and $\text{UAV}_{i \bmod N}$ are swapped each interval i, with N being the number of UAVs in the network.
In contrast, in the network provisioning and the area exploration scenarios, the nodes are considered to be moving in three-dimensional space using state-of-the-art mobility algorithms [Wie14]. Each node represents a small-scale UAV flying at altitudes up to 60 m and a maximum speed of 20 m/s. The UAVs' mobility behavior is composed of two components: microscopic mobility, which addresses the mobility between the UAVs, and macroscopic mobility strategies, which specifies the locations to which the UAVs travel.
In the network provisioning scenario, the microscopic mobility is realized by the Communication-Aware Potential Fields (CAPF) algorithm [Wie14, God11], in which virtual potential forces (e.g., F_{RSS}) are calculated based on communication performance indicators (e.g., RSS), cf. Figure 8.2. The macroscopic location is inherently determined by the users on the ground. In order to extend the coverage area of the UAVs by mean of relaying when the users move away, a role-based connectivity management scheme (see [Wie14, God12a]) is implemented to dynamically reassign the roles of the individual UAVs, based on link monitoring. Both mobility patterns are implemented in the experimental UAV testbed [God12b], which is utilized to generate the mobility traces that is used in the performance evaluation. The impact of the mobility on the link breakage rate μ in this scenario is depicted in Table 8.5. A link is considered non-reliable or broken if its RSS value is below −83 dBm. As the table shows, the stability of the links in case of 16 UAVs in a 3 km^2 area is higher than that of 8 UAVs in a 2 km^2 area because in the latter case the degree of connectivity and overlapping transmission ranges is lower.
In the aerosol plume exploration scenario, the distributed dispersion detection algorithm [Beh13] is implemented. The UAVs are used to detect the borderline of an aerosol plume. At the same time, the UAVs maintains their communication links to exchange sensor and telemetry information. The values of μ in this scenario are illustrated in Table 8.5. Although the swarm is coherent at all times, frequent changes in the links' quality occurs, due to the highly dynamic behavior

Table 8.5: Link breakage rate in the network provisioning and area exploration scenarios [Sbe15].

Scenario	**Network size**	**Channel type**	**Link breakage rate μ [Hz]**
Network provisioning	8 UAVs	Free Space	$4.3 \cdot 10^{-4}$
	8 UAVs	Rice	$4.4 \cdot 10^{-3}$
	16 UAVs	Rice	$6.1 \cdot 10^{-4}$
Area exploration	8 UAVs	Rice	$5.9 \cdot 10^{-3}$
	16 UAVS	Rice	$3.2 \cdot 10^{-3}$

of the UAVs. For instance in case of 8 UAVs, approximately one link is broken every eleven seconds: $\frac{1}{\mu \cdot \#Links} = \frac{1000}{5.9 \cdot 15}\ s$. In this regard, the challenge of the routing protocols is to optimally adapt to these changes to avoid packet drops or long delays.

8.3.5 Simulation Results

In the following, the performance of the protocols is first analyzed in the synthetic grid scenario. Afterwards, the UAV-WMN realistic scenarios (i.e., network provisioning and area exploration) are considered. In all scenarios, the mobility, channel, and traffic models described in the previous sub-sections are used. The protocols are configured according to Table 8.6, based on the findings in [Sbe14c, Hiy13]. Here, two periodic intervals are considered, where the lowest interval should lead to a better performance of the proactive protocol BATMANS in highly dynamic topologies.
HWMPS is operated in the hybrid registration mode to always have the best route from all nodes to the gateway (the ground station) and vice versa. The simulation time of the grid scenario is 300 s. The simulation time of the network provisioning and area exploration scenarios is 900 s. 30 runs are executed in each case, and a confidence interval of 97.5 % is used.

8.3.5.1 Results of the Abstract Grid Scenario

Figure 8.6 depicts the Packet Delivery Ratio (PDR) of the protocols in the grid scenario. Both the route discovery time and the message overhead significantly

Table 8.6: Relevant configurations of the routing protocols [Sbe15].

Parameter	**Protocol(s)**	**Value** [s]
OGM-interval	BATMANS	0.5, 1
Hello-interval	PASER	2, 4
PREQ-interval	HWMPS	2, 4
Purge-timeout	BATMANS	5
Neighbor-hold-time	PASER, ARAN, HWMPS	12
Route-hold-time	PASER, ARAN, HWMPS	15

influence the performance in this scenario, cf. Figure 8.5 (top). HWMPS achieves the best performance, followed by PASER, regardless of the periodic interval's configuration and the Position Swap Interval (PSI), see Figure 8.4. This is justified by the better composition of the delay and the overhead of both protocols: It is shown in Section 8.1 that HWMPS and PASER have a more efficient and robust route discovery process than ARAN and BATMANS, and in this scenario, they also have a lower overhead as $\mu \leq \frac{14}{8} \cdot \frac{1}{300}$ Hz, $\alpha = 1$, $L = 1.5$, $N = 9$, and $\Delta \approx 4.3$, refer to Table 8.4 for the definition of these parameters. Moreover, Figure 8.4 depicts that in case of a Rician channel, the relative performance of all the protocols but that of ARAN is nearly the same. Figure 8.5 shows that the maximum PDR in that case is mainly below 80 %. Due to fading, the topology (i.e., hidden nodes), and the simulation configuration, many channel errors occur in the Rician channel, which lead to retransmissions of unicast frames, thereby, to more collisions and transmission errors. Here, the better performance of HWMPS and PASER than BATMANS, in contrast to ARAN, attests the efficiency of both protocols, as HWMPS, PASER and ARAN are traffic-aware (i.e., they implement a link layer feedback mechanism), while BATMANS is not.

8.3.5.2 Results of the Network Provisioning Scenario

In the network provisioning scenario, Figure 8.7 shows that PASER outperforms HWMPS and ARAN. Besides, the performance of PASER gets better in the longer chain of 16 UAVs in case of the Rician channel. As the route discovery delay of PASER is higher than that of HWMPS (cf. Section 8.1), and it increases the longer the chain is, the results in Figure 8.7 lead to two interpretations: First, the overhead of PASER is lower than that of HWMPS in this scenario. This holds as $\alpha = 1$, $\Delta \approx 2$, and $\mu < \frac{t_p}{2(L-1)}$. For instance, in case of 8 UAVs, $L = 4$, and $\mu_{FreeSpace} < \mu_{Rice} < \frac{1}{4 \cdot 2 \cdot (4-1)}$ Hz, cf. Table 8.5. Second,

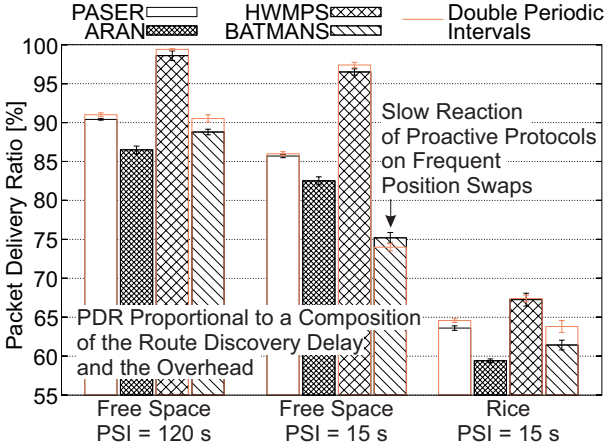

Figure 8.4: Average PDR (PSI: Position Swap Interval) [Sbe15].

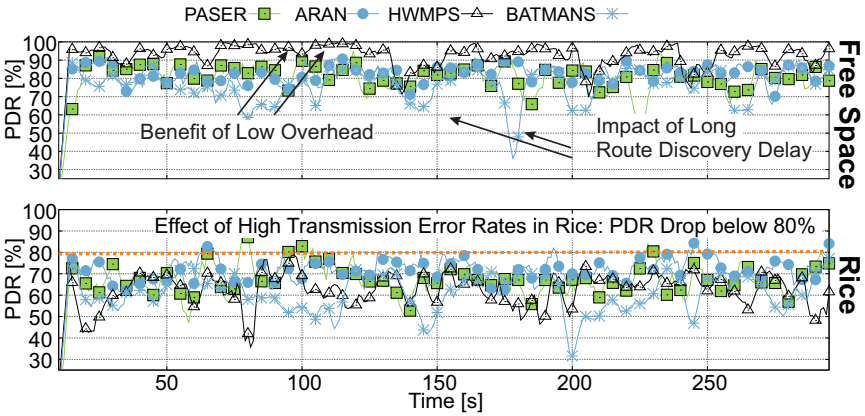

Figure 8.5: PDR vs. Time, PSI = 15 s [Sbe15].

Figure 8.6: PDR in the synthetic grid scenario of nine nodes [Sbe15].

the chain of 16 UAVs is more stable than that of 8 UAVs. This is true since $\mu_{8-Rice} > \mu_{16-Rice}$ (cf. Table 8.5), and the overhead of PASER in the latter case is lower as $\frac{\mu_{8-Rice}}{\mu_{16-Rice}} > \frac{L_{16}^2 \cdot 16}{L_8^2 \cdot 8}$ while α and Δ are mainly the same. Apart

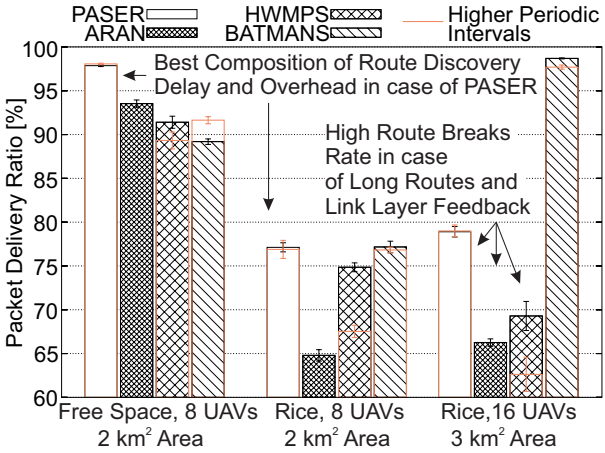

Figure 8.7: Average PDR in the network provisioning scenario [Sbe15].

from these observations, Figure 8.7 reflects that the longer the route is, and the lower the relative mobility is (i.e., in case of 16 UAVs), the better is the performance of the proactive protocol BATMANS. In that case, the probability of route breaks in case of the reactive protocols is higher. This explains the use of proactive protocols in large community networks.

8.3.5.3 Results of the Area Exploration Scenario

Due to the high node density in the area exploration scenario, it is the worst-case scenario for PASER. In contrast to the other protocols, PASER fulfills the *neighbor authentication* goal, and it uses, among others, *hello* messages to maintain this goal. For instance, due to the position information in the *hello* messages, when an authenticated one-hop source moves away, and a wormhole attack is mounted at the new location, the destination would detect the attack upon receiving a *hello* message. Otherwise, the nodes would fall in the attacker's trap until the route get lost, due to collision or timeout. The size of the *hello* messages is proportional to Δ as these messages include information about the one-hop neighbors. In this scenario $\Delta \approx N$ since all the nodes are most of the time one-hop neighbors, $D \approx L = 1$. Despite its *hello* messages overhead, Figure 8.8 shows that PASER can achieve a comparable performance to that of HWMPS in this scenario, given the periodic interval is appropriately set. In contrast, the

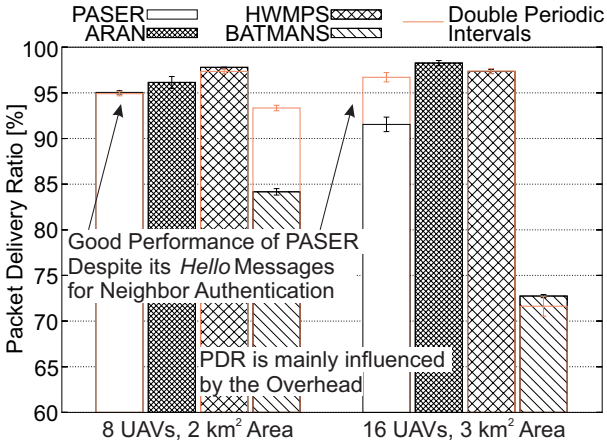

Figure 8.8: Average PDR in the aerosol exploration scenario —Rice [Sbe15].

proactive protocol BATMANS fails to compete in this scenario, especially, at high densities (i.e., 16 UAVs). As the value of μ is below 10^{-2} Hz in this scenario (cf. Table 8.5), the overhead of BATMAN is much higher than that of the other protocols, regardless of the periodic interval's configuration.

The results in this chapter demonstrate that PASER has —in the investigated scenarios (realistic UAV-WMN)— a comparable performance with the well-established, none-secure routing protocol HWMP combined with the IEEE 802.11s security mechanisms. It is also shown that the route discovery process of PASER is efficient and robust and that PASER is scalable with respect to network size and traffic load.

9

Conclusion

This thesis has given an account of the Position-Aware, Secure, and Efficient Mesh Routing approach (PASER). The proposed approach aims to efficiently secure the routing process in highly dynamic WMN in general, in UAV-WMN in particular. To this end, it achieves the security goals: **message authentication**, **message freshness**, **neighbor authentication**, **origin authentication**, and **dynamic key management**. The novelty of PASER is threefold:

1. **Hybrid security scheme:** PASER implements a hybrid cryptosystem, as opposed to the majority of the existing proposals for secure routing in WMNs [Sbe15, Sen13, Sgo13]. Asymmetric cryptography is used for initial mutual authentication and key exchange, after which symmetric cryptography is applied to authenticate the routing messages. Noteworthy with this respect are the two security levels of the symmetric scheme. The first level is based on a group transient key using *MAC*. The second level is based on one-time neighbor authentication tokens using a Merkle tree. Thereby, even if the attacker compromises the group key and eavesdrops all the messages, the attacker can neither impersonate a legitimate node nor fabricate routing messages as the attacker cannot generate new authentication tokens. In contrast, when using the IEEE 802.11s/i security frameworks or symmetric key-based secure routing protocols, once the attacker reveals the key, the attacker can act as a legitimate node. It is shown in Chapter 5 that the computational costs of the PASER symmetric scheme is less than 341 μs on a representative embedded system while RSA-1024 and EDCSA-160 take longer than 26 ms. This combination of values in PASER (security and performance) is deemed to be necessary by the IETF KARP group [IETa] to drive a broad deployment of a secure routing protocol.

2. **In-band key management method:** PASER incorporates an in-band key management method to tackle the interdependency cycle problem between secure routing protocols and key distribution methods [Bob03]. For instance, the majority of the existing proposals for secure routing in WMNs, especially, symmetric-key-based ones and the key management method of the IEEE 802.11s/i security frameworks suffer from this problem [Sbe15,

Zha13b]. In contrast, PASER tackles this issue, allowing for a rapid response to security breaches. This resolves a major issue in current deployments [Leb12].

3. **Specificity to UAV-WMN:** PASER exploits the specifics of UAV-WMN, e.g., the network is operated by one organization, and there is a central unit (the ground station), thus, it is appropriate to deploy a PKI and a KDC. Besides, PASER combats a wide range of routing attacks as it aims to fulfill all the secure routing requirements in UAV-WMN, which were elicited from several research projects, such as [AIR, AVI]. In contrast, the IEEE 802.11s/i and the majority of the existing proposals for secure routing in WMNs are vulnerable to the wormhole or the blackhole attacks, see [Sbe14b, Sen13, Sgo13, Abu08].

The main conclusions of this thesis can be summarized as follows.

- **Validation of the WMN model in INETMANET-OMNeT++:** The WMN model in INETMANET-OMNeT++ has been validated based on theoretical analysis and experimental measurements. It is shown that different WMN routing philosophies and protocol parametrization lead to similar performance results in simulation and in practice, i.e., to valid conclusions, if the simulated network is appropriately configured.

- **Implementation of PASER in INETMANET-OMNeT++:** A modular implementation of PASER is contributed to the INETMANET framework of OMNeT++. This easy-to-adapt implementation has been recently integrated in the official INETMANET framework as the first implementation of a secure routing protocol in OMNeT++, see [INE13].

- **Design, implementation, and evaluation of the generic Linux kernel framework ROUTE-O-MATIC:** The ROUTE-O-MATIC framework is proposed, and its implementation design and evaluation are discussed. ROUTE-O-MATIC provides special features for reactive routing protocols for which the current network subsystem of Linux is not designed. Reactive routing logic mainly relies on two core features of the underlying operating system:

  1. A mechanism which notifies the routing logic in case a route to an unknown destination is needed;

  2. A buffer which temporarily saves the packets for the unknown destination while the route discovery is performed and re-injects those after establishing the route.

  The network subsystem of the Linux operating system, which is the default platform for the implementation of WMNs, lacks of support for both

features. ROUTE-O-MATIC enables these features, it is ready-to-use and provides a level of completeness to run any routing logic. ROUTE-O-MATIC is evaluated in different scenarios to appraise its time overhead, which is shown to be negligible. For instance, ROUTE-O-MATIC generally adds an overhead of less than 5.5 μs to the processing time of a routable packet. Compared to the Linux kernel processing time, this is only an increase of less than 5 %. ROUTE-O-MATIC is used to implement PASER in Linux. Its code is online available on *www.paser.info*.

- **Implementation of PASER in practice:** The feasibility of the PASER approach and its benefits have been presented in different events and international conferences. These include, but are not limited to, the Vodafone innovation days [Sbe14a], the research work on the security and routing in wireless mesh networks [Sbe14b], and the early demonstrator of the SecInCoRe research project [Kuh15, SEC]. The PASER experimental code is open source, and it is available on the official website *www.paser.info*. To the best of the author's knowledge, the PASER experimental implementation is the only online available up-to-date reference for a secure WMN routing proposal.
- **Experimental analysis of the blackhole and wormhole attacks:** The robustness of the PASER and the IEEE 802.11s/i against the blackhole and wormhole attacks are investigated in a testbed. It is demonstrated that the IEEE 802.11 security frameworks are not able to mitigate both attacks, thus, these cannot be deployed in critical scenarios, e.g., in UAV-WMN. It is concluded that using these security frameworks in the backbone of WMNs is not as appropriate as their conventional use to secure the communication between mesh access points and clients. In this regard, an efficient secure routing protocol combined with a dynamic key management scheme are inevitable to establish a reliable network, as experimentally shown using PASER. It is also shown that PASER mitigates —in UAV-WMN— more attacks than the well-known, secure routing protocol ARAN, and its key management is more fail-safe than that of ARAN.
- **Theoretical and simulation-based analysis of the route discovery delay and the message overhead:** The route discovery delay of PASER and three alternatives is analyzed in theory and in simulation. Lower bound equations of this delay are derived as it constitutes along with the message overhead, for which asymptotic expressions are provided, the main impact on the overall network performance. The results show that PASER has an efficient and robust route discovery process, and it is scalable with respect to network size and traffic load.
- **Simulation-based performance evaluation in realistic UAV-WMN:** Using the network simulator OMNeT++, realistic mobility patterns of

UAVs, and an experimentally derived channel model of the air-to-air link between the UAVs, it is demonstrated that in UAV-WMN-assisted network provisioning and area exploration scenarios PASER has a comparable performance with that of the well-established, none-secure routing protocol HWMP combined with the IEEE 802.11s security mechanisms.

10

Directions for Future Research

Contents of this Chapter

The work performed in this thesis can be extended in various directions. Investigating the use of PASER in a broader range of application scenarios is one direction. Another compelling direction is the virtual localization of position-unaware nodes. This enables a geographical leash-based mitigation of the wormhole attack in scenarios where satellite services are not constantly available. A first approach in this direction is proposed in the following. Besides, further ideas to enhance the security of PASER and to optimize its performance as well as new research avenues are discussed.

Publications. Parts of this chapter have been presented in [Sbe12a].

10.1 Virtual Localization Extension for Geographical Leashes

The secure wireless mesh routing protocol PASER, proposed in this thesis, uses geographical leashes (satellite-based position information) to combat wormhole attacks in outdoor scenarios with low obstacles, e.g., in UAV-WMN. In case

of many obstacles or indoor scenarios, satellite services are however not available. Thereby, legitimate nodes cannot participate in the protocol, and they are excluded from the network. This contradicts with the design goals of WMNs, especially, 'ubiquitous access anywhere'. To address this issue, a novel Virtual Localization Extension (VLX) for the localization and integration of position-unaware nodes with the rest of the network is introduced in this section. The use of VLX is not restricted to PASER, it is a generic extension for geographical leashes in scenarios in which nodes are not aware of their position (e.g., unmanned ground vehicles indoors).

10.1.1 Review of Countermeasures Against the Wormhole Attack

The necessity of countermeasures against the wormhole attack is beyond dispute in the literature [Kha09]. To mitigate this attack, the authors in [Cap03] propose the SECure Tracking Of node encounteRs (SECTOR) protocol. Their approach is based on a special hardware, which sends 1-bit-RREQs that must be immediately answered by the corresponding receiver. Here, the messages do not enter the CPU path, instead, they are processed with very low latency by the special hardware. Attackers cannot easily imitate this behavior due to the delay imposed by the wormhole tunnel. This approach, however, requires the integration of new hardware in the mesh nodes while yielding positive results only if the challenge-messages do not collide. Otherwise, this approach could produce false alarms. Thereby, the rate of profit by deploying it on a wide range of devices is perceived to be low. Two other examples that fall into the category of the route delay analysis are the Delay Per Hop Indication (DelPHI) [Chi06] and the Wormhole Attack Prevention (WAP) [Cho08] protocols. Both identify affected routes by measuring either packet travel or round trip time per hop. Even though the implementation of both approaches is straightforward and does not require extra hardware, only balanced networks offer a good basis for such comparisons. In networks with frequent topology changes or where the link quality often varies, the aforementioned metrics rather produce false alarms.
Apart from that, the authors in [Kha05] propose the LiteWORP protocol. Here, nodes have to be placed with largely overlapping reception ranges. Each node has to be operated in promiscuous mode, so it can check if other nodes forward packets as expected. As soon as anomalies with respect to packet delivery ratio or delay are detected, nodes causing these anomalies are completely excluded from any further communication. The requirement for overlapping reception ranges in real life WMN applications is however not always guaranteed, especially, in indoor scenarios. Besides, the overlapping reception ranges could yield to a high interference ratio, thereby, to a deterioration in the network performance.
A different method to combat the wormhole attack is proposed in Castor [Gal10], which aims to provide secure routing without the need for routing messages (see

Section 3.2). This approach is both simple and protects against the wormhole attack. However, it produces a considerable overhead due to the header added to each data packet and because of the acknowledgment message required from each destination for each packet. In addition, it assumes that each pair of end nodes either shares a symmetric key or they know the public key of each other. This is, however, not straightforward in WMNs, as discussed in Section 3.2.

10.1.2 Review of Indoor Localization Schemes

Most existing localization schemes are based upon the following techniques [Lew12]: Time Difference of Arrival (TDoA), Time of Arrival (ToA), Angle of Arrival (AoA), Cell of Origin (CoO) / Cell-ID or Trilateration by RSS. All these techniques but the last one are only feasible in specific scenarios. For instance, both ToA and TDoA require tight clock synchronization. AoA is based on special antenna arrays which are rarely incorporated in mobile nodes. The Cell-ID is only feasible in networks with several stationary base stations. In contrast, RSS values are available in most situations and easily accessible in Linux-based operating systems. Yet, these values are not always accurate, especially, in indoor scenarios. Thereby, range-free localization techniques based on non-accurate RSS are of interest in this research. The Approximate Point In Triangle Test (APIT) [He03] falls into this category. For a given set of position-aware anchor nodes N and a position-unaware node I, all $\binom{3}{N}$ possible triangles are calculated to test whether I is inside those triangles or not, based on a comparison of RSS. For each positive test, the corresponding triangle is written into an APIT map. Finally, the center of gravity is calculated for this map. Since individual tests are rather coarse, this final step significantly increases the quality of the localization. Nevertheless, this approach does not only require that anchor nodes are in the proximity of I, but it also assumes that those nodes even surround I, which does not always hold in mobile scenarios. Besides, this approach requires a higher anchor node density than its counterpart, the Centroid algorithm [Bul00]. According to the latter, the approximate position of the node I, P'_I is estimated by calculating the centroid of the sum of the nodes in the adjacencies of I using Equation 10.1.

$$P'_I(x,y) = \frac{\sum_{j=1}^{N} P_j(x,y)}{N} \tag{10.1}$$

(x, y) are the coordinates, P_j is the position of the nodes in adjacencies, while N is their total number. A more sophisticated approach in this context is to give each point an individual weight that reflects the values of the RSS as described

in [Blu07]. Thereby, the position of the node I is now given by Equation 10.2.

$$P'_I(x,y) = \frac{\sum_{j=1}^{N} (w_{Ij} \cdot P_j(x,y))}{\sum_{j=1}^{N} w_{Ij}} \tag{10.2}$$

w_{Ij} is the weight of the relation of nodes I and j. This approach is termed Weighted Centroid Algorithm. It is robust against RSS fluctuations, and a minimum of only one node is necessary for the algorithm to work properly. Nevertheless, a slight drawback of it is that trilateration only resides within the convex hull spanned by the points P_j. While this is not a problem in scenarios with random distribution of nodes with/without position information, it can heavily distort the localization in networks where many nodes without position information are physically isolated. This makes the approach vulnerable to the wormhole attack since in that case it significantly modifies the topology of the network. However, due to the many aforementioned advantages it endorses, this approach is partly adopted in VLX.

10.1.3 Requirements and Goals for Virtual Localization

The VLX is designed with the following requirements in mind:

- *Conservation of resources:* Except for the gateway, batteries of mobile nodes are limited in capacity and should not be consumed any further than necessary. As a consequence, VLX overhead and usage of CPU should be kept to a minimum.
- *Changes of the existing security level:* The existing security level of the routing protocol should by no means be restricted. The VLX is an extension to increase the security indoors. That is, all nodes with a valid position information must adhere to the present mechanisms. Only nodes without position information may deviate from this behavior.
- *Mapping of the network topology:* In order to keep use of geographical leashes in case of obstacles or indoors, the virtual projection of the real topology of the network must be as accurate as possible. Bearing this in mind, two potential mistakes might occur. A node is considered as a neighbor in the virtual world although it is not in the real world (alpha error), or it is not a neighbor in the virtual world while it is in the real world (beta error). For a proper protection against the wormhole attack, the alpha error must never occur, while the beta error is not that critical, nevertheless, it must be kept as low as possible.

- *Accuracy of the localization of nodes:* Requirements for the accuracy of the localization using VLX are rather low. The localization just has to be accurate enough to ascertain the correctness of the topology. Apart from that, accuracy is a secondary optimization problem.

Central Goal: The main goal for the design of the VLX is formally defined as follows: For all nodes $n_i = \{n_1, n_2, \ldots, n_N\}$ in the direct adjacencies $\mathbb{A}$ of a node I, the following must always be true:

$$n_i \varepsilon \mathbb{A}_{virtual}(I) \Rightarrow n_i \varepsilon \mathbb{A}_{real}(I) \qquad \text{for } \forall t \geq 0 \tag{10.3}$$

The set of adjacent neighbors $\mathbb{A}_{virtual}$ is therefore a subset of the set of real neighbors $\mathbb{A}_{real}$, such that:

$$\mathbb{A}_{virtual}(I) \subseteq \mathbb{A}_{real}(I) \qquad \text{for } \forall t \geq 0 \tag{10.4}$$

As a result, the following must always hold:

$$\sum(n_i | n_i \varepsilon \mathbb{A}_{virtual}(I)) \leq \sum(n_i | n_i \varepsilon \mathbb{A}_{real}(I)) \qquad \text{for } \forall t \geq 0 \tag{10.5}$$

10.1.4 The Virtual Localization Extension Approach

In WMNs running routing protocols such as PASER, where the nodes register themselves at a central unit, e.g., the KDC, this unit knows about the positions of all the nodes that are aware of their position information. This information is used in VLX to localize a position-unaware node as well as to give it a hint on which nodes to trust as one-hop neighbors.
While in most geographical leash-based mechanisms, a perfect unit circle [Cla90] is an appropriate paradigm to determine one-hop neighbors, the assumed fixed maximal transmission radius is not applicable in indoor environments. For this reason, it is proposed in VLX to define the neighborhood/range of a node based on the usage of an ellipse as a generalization of a circle. The ellipse brings a lot of advantages over the standard circle while conserving the relevant characteristics of the latter. Both have in common a geometric shape that forms a convex boundary. Both can be shrunk and shifted. Thus, both are very useful to enclose the one-hop neighbors scattered around a node. However, in contrast to the circle, the ellipse might also be transformed so as to approximate a line. On top of that, it might be rotated, offering in combination with the transformation characteristic a great flexibility to selectively enclose one-hop neighbors in the adjacencies. In fact, this is an indispensable feature for fulfilling the VLX goal given in Equation 10.4.

In its simplest form and without any rotation, the definition of an ellipse is very lightweight. It is defined by only four scalars in the Cartesian plane, namely by minor and major axis (a_{min}, a_{maj}), and coordinates (x_c, y_c) of a center point c. The corresponding equation of the ellipse in the Cartesian plane is given by:

$$\frac{1}{a_{maj}}(x - x_c)^2 + \frac{1}{a_{min}}(y - y_c)^2 = 1 \tag{10.6}$$

In its sophisticated form, an ellipse, a conic section with arbitrary size, shift, and rotation, is defined as the set of points $P_{ni} = (x, y)$ which satisfies the following equation:

$$rx^2 + sy^2 + txy + ux + vy + w = 0 \tag{10.7}$$

For performance reasons, VLX exploits the center form representation [Gär97] in combination with the position shifting $\Delta = (\Delta x/\Delta y)$ to define an ellipse. In this case, only five scalars $r, s, t, \Delta x$, and Δy are required. This paradigm reduces the point-in-ellipse verification test to a small number of multiplications and summations. Here, a given point P_{ni} is located inside an ellipse defined by the set of parameters M and the center point c if and only if the following holds:

$$(P_{ni} - c)^T M (P_{ni} - c) - 1 \leq 0, \text{ with } M \in \Re^{(2x2)} = \begin{pmatrix} r & t \\ t & s \end{pmatrix} \tag{10.8}$$

To localize an indoor node I using VLX, the following five steps are taken:

- **Step 1**: One-hop neighbors append the RSS values of I's RREQ packets to those packets.
- **Step 2**: Upon receiving those packets, the central unit (i.e., gateway or KDC) derives the network propagation direction towards I.
- **Step 3**: The central unit runs Algorithm 1 to virtually localize I.
- **Step 4**: The central unit runs Algorithm 2 to generate the ellipse, which represents the neighborhood/range of I.
- **Step 5**: The central unit appends the virtual position of I and its ellipse to each route reply message addressed to that node.

To ease the understanding of these steps, the example given in Figure 10.1 is used. In this example, an indoor node I wants to join the network. Thereby, it triggers a registration at a gateway.

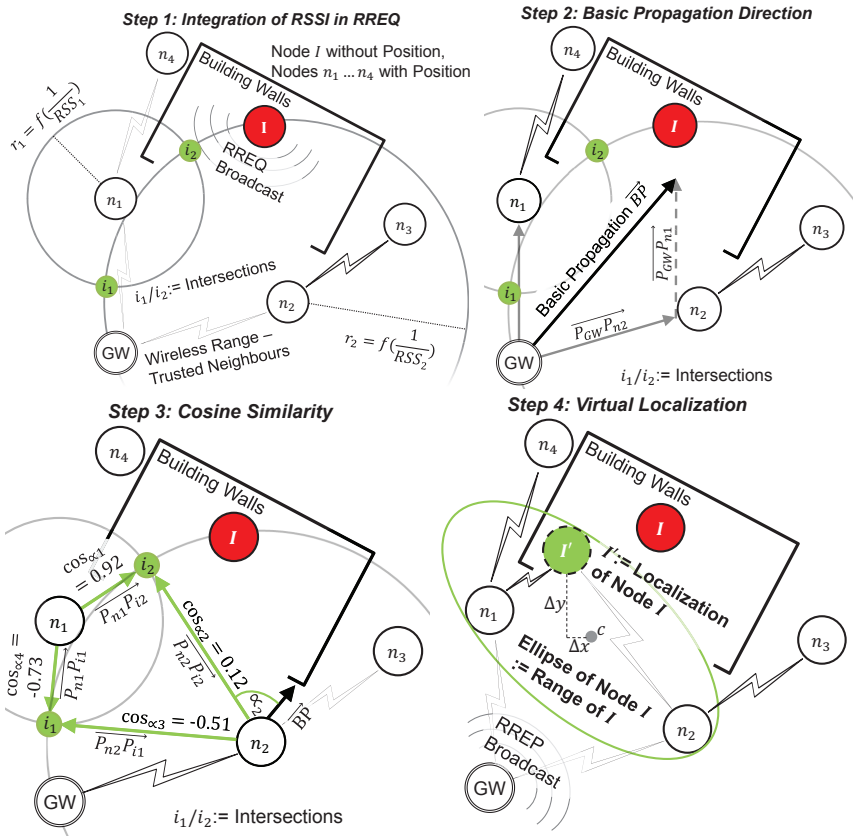

Figure 10.1: Overview of the main steps to virtually localize a node using VLX [Sbe12a].

Step 1: Integration of RSS Values in RREQ Packets. When running secure mesh protocols such as PASER, a node wanting to join the network typically starts a registration process at the KDC, mainly to grant access to the network. This node broadcasts a route request towards a gateway to reach the KDC. This process is illustrated in Figure 10.1 (top, left). I broadcasts a RREQ message looking for a gateway. While RREQ messages typically must include the position information of a node, by deploying VLX, those messages are extended by one flag, which position-unaware nodes set to indicate that they have not yet gotten

valid localization data. Upon receiving such RREQ messages, one-hop neighbors forward the RSS values of these messages to the next-hop towards the gateway. From this stage, the message processing and forwarding logic did not change. That is, any node beyond one-hop distance forwards these messages as usual.

Step 2: Calculation of Network Propagation Direction. The direction in which the network propagated to reach I is calculated at the central unit (i.e., KDC or gateway). It is derived from the position of all one-hop neighbors that forwarded RREQ messages of I to the gateway. This direction is calculated by averaging the vectors between the gateway and those neighbors. That is, the basic propagation is obtained by summing the normalized vectors from the position of the gateway P_{GW} to the position of I's one-hop neighbors, P_{ni}, e.g., P_{n1} and P_{n2} in Figure 10.1 (top, right). Thus, the basic propagation $\overrightarrow{BP_I}$ towards a node I with $n_i \varepsilon \mathbb{A}_{virtual}(I)$ is given by:

$$\overrightarrow{BP_I} = \frac{\overrightarrow{P_{GW}P_{n1}}}{\left\|\overrightarrow{P_{GW}P_{n1}}\right\|} + \frac{\overrightarrow{P_{GW}P_{n2}}}{\left\|\overrightarrow{P_{GW}P_{n2}}\right\|} + \cdots + \frac{\overrightarrow{P_{GW}P_{n_{max}}}}{\left\|\overrightarrow{P_{GW}P_{n_{max}}}\right\|} \tag{10.9}$$

Step 3: Virtual Localization of Position-Unaware Nodes. The localization of a position-unaware node I is carried out at the central unit as specified in Algorithm 1. Hereby, two mechanisms are mainly used. In scenarios where the position-unaware node has at least three position-aware one-hop neighbors, the Weighted Centroid Algorithm (WCA) is used. In scenarios where the position-unaware node has less than three position-aware neighbors, a novel approach based on the combination of RSS lateration and the cosine similarity is applied. The reason for incorporating this approach in VLX lies in the high topology mismatches/collisions caused by the WCA in the aforementioned case. Laterations of the received RSS values are used to calculate all intersections i of the circles with the center set to the one-hop neighbors' positions and the radius chosen as a reciprocal of the corresponding RSS values, as illustrated in Figure 10.1 (bottom, left). In that case, because of RSS fluctuations and imperfect channel characteristics, it is likely to have more than one perfect intersection. This is where the novel cosine similarity approach comes into play. This approach is adopted from the text analysis field where it is used to analyze the analogy of vocabulary. As for VLX, the cosine of the angles between the basic propagation $\overrightarrow{BP_I}$ towards a node I and the vectors between the gateway and the intersection points $\overrightarrow{P_{GW}P_i}$ is calculated. Hence, the definition of the cosine similarity in this context is given by the equation:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}} \tag{10.10}$$

Algorithm 1 Localization of a position-unaware node I (run by a central unit)
Input: Map of RSS values
Output: Virtual coordinates $P'_I = (x, y)$

```
list ℕ ← get all N entries in RSS map with position information
if ∑ℕ = 0 then
    list ℕ ← get all N entries in RSS map
else if ∑ℕ = 1 then
    B̂P ← calculate normalized Basic Propagation
    dist ← estimated distance                          ▷ based on f(1/RSS)
    P'_I ← P_ni + (B̂P · dist)
else if ∑ℕ = 2 then
    B̂P ← calculate normalized Basic Propagation
    dist_1/2 ← estimated distance                      ▷ based on f(1/RSS_1/2)
    get all intersections i_1/i_2                      ▷ ∑ i_n ε 0..2
    if ∑i = 1 then
        P'_I ← P_i1
    else if ∑i = 2 then
        get cosineSimilarity for all P_n P_i
        P'_I ← P_i | P_i with MAX cosineSimilarity
    else
        call Weighted Centroid Algorithm (WCA):
        P'_I = (P_n1 · RSS_1 + P_n2 · RSS_2) / ∑RSS_n
    end if
else                                                   ▷ ∑N ≥ 3
    call Weighted Centroid Algorithm (WCA)
end if
```

where n is the dimension of the vector space. The codomain of the cosine similarity $\mathbb{W} = [-1, 1]$ either indicates an exact match of the angle (1), orthogonality (0), diametrically aligned angles (-1), or values in between. In VLX, the intersection point with the highest cosine value is chosen as an appropriate position for the indoor node. Figure 10.1 (bottom, left) depicts an example of this approach.

Step 4: Calculation of the Enveloping Ellipse. RREQ messages are received by the central unit (gateway or KDC) time-discretely. That is, the number of available one-hop neighbors of a position-unaware node I and their corresponding RSS values continuously increases. From the time the first value is available, the central unit calculates both the estimated position and the corresponding ellipse for I and answers the request by a RREP message including this information. Hereby, the central unit updates those information by each incoming request/outgoing reply. To achieve the goal in Equation 10.4, only the one-hop neighbors from which the central unit already received a RREQ must be located in the ellipse of I. This is rarely given by an ellipse centered at the virtually localized position P'_I. In most cases, a shifting of the ellipse is necessary to guarantee the topology's integrity without disproportional costs. Figure 10.1 (bottom,

Algorithm 2 Calculation of an enveloping ellipse (run by a central unit)
Input: Position of nodes n_i in $\mathbb{A}_{real}(I)$
Output: Ellipse parameter set $r, s, t, \Delta x, \Delta y$ for $\mathbb{A}_{virtual}(I)$

```
N ← ∑ entries e ∈ RSS map
runNumber ← 0
ellipseIsColliding ← true
while ellipseIsColliding do
    availableRSSEntries a ← N - runNumber
    runNumber ← runNumber + 1
    ellipse set r, s, t, Δx, Δy ← call calculateEllipse(top a, entries e)
                                  ▷ VLX checks P_n1P_n3 − α · P_n2P_n3 ≠ 0   ∀α ∈ ℜ
                                  ▷ and ensures ≥ 3 linearly independent positions
    ellipseIsColliding ← call checkForCollision(r, s, t, Δx, Δy, e)
                                  ▷ for ∀ entries e | e ∈ routing table ∧ e ∉ RSS map
end while
```

right) gives an example of such a scenario. Here, the center of the ellipse is shifted by Δ. That is $\Delta = (\Delta x/\Delta y)$ is added to the node's virtual position $P'_I = (x_I, y_I)$ to get the ellipse's center point $c = (x_c/y_c)$. Algorithm 2 specifies all the steps a central unit undergoes to compute the ellipse. The basic idea of this algorithm is to recalculate the ellipse as long as the point-in-ellipse verification test is positive for any node n_i which is not in the direct proximity $\mathbb{A}_{virtual}(I)$, but whose position P_{ni} is inside the ellipse of I. That is, this node collides with the ellipse area. In this case, the ellipse is altered to exclude that node. The algorithm ensures that the central goal for protecting the network topology from aberrations is always fulfilled. Despite its simplicity, the algorithm is an important pillar of the entire concept.

Step 5: Transmission of Virtual Position and Information about Adjacencies (Ellipse). Instead of transmitting real position information, a position-aware node I sends its virtual position $P'_I = (x_I, y_I)$ and information about its adjacencies using the set of parameters that define its ellipse, namely, $r, s, t, \Delta x, \Delta y$. The node receives this information during the registration phase from the replies it gets from the KDC or the gateway.

10.1.5 Selected Performance Results

The proposed VLX approach was integrated in PASER and evaluated in simulation in different indoor/outdoor scenarios and with different nodes constellations, see [Sbe12a]. The results showed that VLX has negligible to no side effect in scenarios where position information are available. In scenarios where position information are not available, VLX enables legitimate position-unaware nodes to access the network. It provides nearly the same level of security as geographical

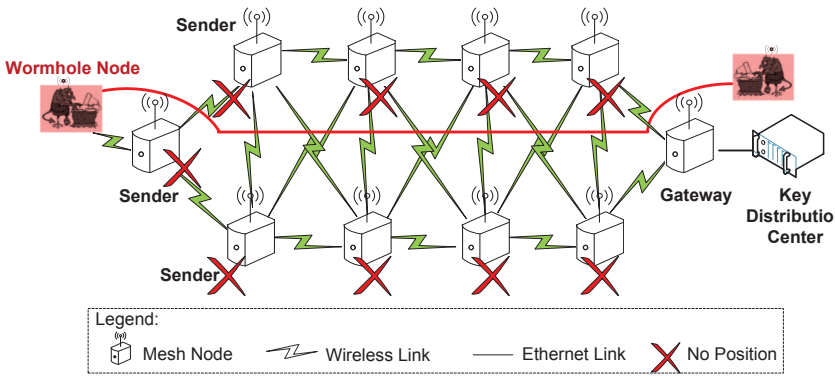

Figure 10.2: Network setup to evaluate the VLX ratio sensitivity of position-unaware to position-aware nodes [Sbe12a].

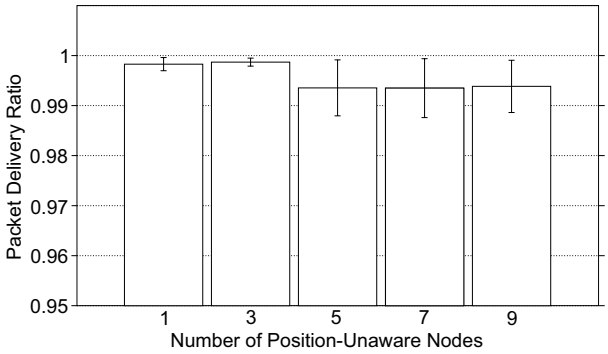

Figure 10.3: Results of the ratio sensitivity of position-unaware to position-aware nodes using PASER with VLX [Sbe12a].

leashes in outdoor scenarios with no obstacles, it has a light overhead, and it is not sensitive to low ratios of position-aware to position-unaware nodes, as illustrated in Figure 10.3. Here, the performance of PASER with VLX is evaluated by simulating an increasing number of nodes without position information in a network of ten nodes under wormhole attack, using the simulation configurations of Table 10.1. Nodes were increasingly set to work without position information from the left-hand side to the right-hand side, as depicted in Figure 10.2. At the beginning, the ratio of position-unaware to position-aware nodes was set to 1/9.

Table 10.1: Relevant simulation configurations [Sbe12a].

parameter	Value	Parameter	Value
Simulation tool	OMNeT++	Simulation time [s]	625
Channel model	Free-Space	Mac layer	802.11a
Mobility pattern	Static	Traffic model	CBR-UDP
Number of simulations	10	Packet size [Byte]	512
Number of traffic flows	3	Data rate [kbit/s]	512

Afterwards, the number of nodes without position information was successively increased by two to finally end at an opposite ratio of 9/1 position-unaware to position-aware nodes with only the gateway having position information (which is typically true in most real-life scenarios). As Figure 10.3 shows, it is apparent that the PDR remains nearly constant in average regardless of the considered ratio. The mean values differ by less than 1 %. In other words, this figure reveals that VLX is suitable for critical scenarios with bad satellite-service. The protection remains active throughout the whole network lifetime so that wormhole links are always avoided.

10.1.6 Open Issues

While VLX is very effective in case the mesh nodes are static and the wormhole attack is mounted after deploying the network, it is perceived that this approach is ineffective in case the wormhole attack is active before setting up the network. In the latter case, the information received at the gateway are corrupted, and this leads to wrong conclusions (corrupted virtual positions). Provided a satisfactory number of nodes that are not affected by the wormhole attack, this issue can be solved using statistical means to detect the corrupted information. A remaining issue is however the use of VLX in mobile scenarios, which are not considered in the VLX design. This is an interesting challenge for future work, especially, with respect to latency as a new virtual positions must be frequently calculated.

10.2 Further Directions for Future Research

- **Formal security analysis of PASER using model checking tools:** In addition to PASER, a profusion of secure routing protocols have been proposed in the literature to establish reliable routes in MANET and WMNs,

see [Abu08]. However, in most of these proposals, non-exhaustive methods are used to evaluate their security, see [Sen13, Sgo13, And07]. This is also the case in PASER, even though, visual inspection, simulations, and experiments are used to be as exhaustive as possible. Non-exhaustive security verification methods could lead to a false sense of security since many of these 'secure' protocols are later shown to be vulnerable. Note also that exhaustive methods could also lead to wrong conclusions, as in the case of Ariadne [Hu05], where provable security (exhaustive method) is used to prove the security of the protocol, yet, the protocol is shown later to have several security flaws, [Bur09]. Nevertheless, using exhaustive security verification methods, the security goals are fulfilled with higher probability. Apart from that, the implementations of secure routing protocols are mostly not validated to strictly match the design. This leads to security flaws even if the protocol design is secure. Thereby, in future work, efforts could be devoted to use automated model checking tools (exhaustive method), for instance the COMPLeTe framework [Grö13], to verify the security of both the design and implementation of secure routing protocols, such as that PASER.

- **Trajectory-Aware Routing in UAV-WMN:** The mobility of UAV mesh nodes in UAV-WMN can be predicted to a large extent, either because the trajectory is predefined or because the UAVs implement a well-known mobility strategy, e.g., potential fields [Wie14, God11]. Thereby, mobility-based link breaks and even link deteriorations can be anticipated. By extending UAV-WMN routing protocols to be trajectory-aware and to implement early actions, e.g., switch-before-break, route failures could be avoided, and the performance could be optimized. PASER, which is already supports the exchange of position information between neighbors, builds a suitable basis for such a future work.

- **Physical layer-based key management instead of PKI:** The main security overhead of PASER is caused by the digital signature operations (processing delay) and the exchange of certificates (transmission delay) to establish trust between new one-hop neighbors, after which the efficient PASER symmetric scheme is used. This overhead could be removed if it were possible to implement another scheme to establish the trust. A paradigm that is arising with this respect is physical layer security, see [Gol11] and the references therein. Here, a secret-key is generated between one-hop neighbors, without pre-shared knowledge, based on channel characteristics that only both parties can measure. Even though the feasibility and robustness of this approach in practice has not been proven yet, it is a compelling avenue for future work.

- **Integrating the PASER security scheme in IEEE 802.11s:** It is shown in this thesis that the IEEE 802.11s does not provide sufficient security in WMN backbones. An interesting future direction in this respect is to extend the IEEE 802.11s to include the PASER security scheme, since both PASER and HWMP (the routing protocol specified in IEEE 802.11s) have their routes in AODV. As a first step towards achieving this goal, simulation can be used. Here, the contributed simulation model of the IEEE 802.11s/i to the official INETMANET framework of OMNeT++ builds a good foundation, see [Neh14].
- **Design of a robust and interference-free channel access in WMNs:** One of the main reasons of routing instabilities in WMNs is interference due to CSMA/CA, see [Ng07]. This in one hand leads to performance deterioration. On the other hand, it burdens the deployment of WMNs in several applications, e.g., the Command and Non-Payload Communication (CNPC) link of UAVs in practice, as this link should ensure that the UAVs can always be remotely piloted and their position is known. That is, the use of WMNs in this case is restricted to the payload link of UAVs [FAA]. To address this problem, robust and interference-free channel access needs to be designed for WMNs. A first approach is proposed in [Hie08]. Further work in this respect could strongly impact the breakthrough of WMNs.

A

Brief Introduction to Cryptography

Contents of this Chapter

Cryptography derives from the Greek word *kruptos*, which means hidden. It is defined as the art and science of secret writing [Sch95], where encryption is the process of transforming plaintext into ciphertext, and decryption is the reverse process. The cryptographic algorithm (i.e., mathematical function) used for encryption and decryption is called a cipher. Modern cryptography follows the Kerchkhoffs principle, which states that the security should definitely not rely on the secrecy of the algorithm. Thus, modern ciphers are publicly known, yet they are used in conjunction with keys, and these keys or part of them are kept secret. The cryptographic algorithm, the keys, and all possible plaintexts and ciphertexts build the cryptosystem. The security of the system mainly depends on the strength of the cryptographic algorithm and the secrecy of the key.
A fundamental objective of cryptography is to address the following four security goals [Men96]:

- **Confidentiality:** Keeping the content of information secret for all but those authorized to access it, e.g., by encrypting the information.

- **Integrity:** Protecting the information against accidental or unauthorized alteration. It should be possible for the receiver of information to detect manipulation in transit.

- **Authentication:** This goal can be applied to both the communicating parties and the information. Thus, it covers two concepts: *node authentication* and *message authentication*. *Node authentication* is the ascertaining of the identity of the communicating parties. Unauthorized parties should not be able to masquerade as authorized ones. *Message authentication* denotes origin (node) authentication and the integrity of the information.
- **Non-repudiation.** Preventing a party from denying previous commitments or actions, making repudiation impossible if a dispute arises, e.g., ensuring that the originator of a message cannot deny the creation of the message.

In the following, the common cryptographic algorithms used to achieve these goals are briefly described. They are typically divided into two classes: symmetric-key and public-key algorithms. Table 3.1 depicts an overview of the main characteristics of each class.

A.1 Symmetric-Key Cryptographic Algorithms

Symmetric-key cryptographic algorithms rely on a single key (or several keys calculated from each other). For instance, in the case of ciphering, the same key is mostly used for encryption and decryption, thus, the cipher is appropriately called a symmetric cipher. Symmetric-key cryptosystems have been used for 4000 years [Sim82]. Thereby, they are also termed conventional cryptosystems. They are characterized by being fast, and they rely on relatively short keys to provide long term security. However, they have an inherent problem, which is the distribution of the key to both the sender and the receiver.
Symmetric-key cryptographic techniques are in widespread use, especially to provide mutual authentication as well as data confidentiality and integrity [Paa09].

A.1.1 Symmetric Ciphers

Let p be the plaintext, c the ciphertext, e the encryption key, d the decryption key, and K the key space, i.e, $\{e_1, d_1, ..., e_n, d_n\} \in K$ with $n \in \mathbb{N}$. A cipher is called symmetric if for each key pair (e, d), it is possible to calculate d knowing only e, and vice versa [Men96].
The encryption and decryption functions E and D of a symmetric cipher are defined as follows.

- Encryption: $c = E_e(p)$;
- Decryption: $p = D_d(c)$.

Both functions have the property that $D_d(E_e(p)) = p$. In most practical systems (e.g., WLAN), the same key is used, i.e., $e = d$. Encryption and decryption are then inverse operations. Symmetric ciphers are generally categorized as being either stream ciphers or block ciphers.

Stream Cipher. The same function is used for encryption and decryption, and is applied to each bit individually. Let p_i, c_i, and e_i be the i^{th} bits of the plaintext, ciphertext, and encryption key, respectively. Let $\oplus$ denote an XOR operation, i.e., an addition modulo 2. The encryption and decryption operations of a stream cipher are defined as follows.

- Encryption: $c_i = E_{e_i}(p_i) \equiv p_i \oplus e_i$;
- Decryption: $p_i = E_{e_i}(c_i) \equiv c_i \oplus e_i$.

Stream ciphers are in general faster than block ciphers [Zha05, Tra10]. The generation of a pseudorandom key poses however a major challenge in practice. Well-known examples of stream ciphers are the A5/1 in GSM, RC4 in WLAN (WEP and WPA), and SNOW 3G in LTE. The A5/1 and RC4 are already broken, the SNOW 3G not yet.

Block Cipher. A block cipher splits the plaintext into blocks having a fixed bit length. The whole block is encrypted at a time using the same key, and a ciphertext block of the same length as the plaintext block is generated. Here, the encryption of each bit in the plaintext block depends on every other bit in the same block. The vast majority of network-based symmetric cryptographic applications use block ciphers, such as the Advanced Encryption Standard (AES) [Tra11]. AES is used in the Internet Protocol security (IPsec), Transport Layer Security (TLS), in addition to being the mandatory encryption algorithm for US government applications. It is also used in WLAN and LTE. It was standardized by the US National Institute of Standards and Technology (NIST) in 2001 after a five-year selection process. AES has been intensively studied and no attacks have been found that are better than the brute-force attack (testing all possible keys), against which AES provides long-term security [Paa09]. AES has a block length of 128 bits. In order to encrypt plaintexts of arbitrary length (i.e., to operate as a stream cipher), several modes of operation for block ciphers such as the CounTeR mode (CTR) have been standardized [CTR80]. For instance, the CTR mode is applied to AES in WLANs. To encrypt a plaintext p in the CTR mode, p is split into n 128-bit blocks $p_{b_1}, ..., p_{b_n}$, and n 128-bit counters are generated $cnt_1, ..., cnt_n$. Here, cnt_i = intial value || counter value$_i$, and the counter value is typically incremented by one for each subsequent counter. The cipher text c is the concatenation of the encrypted blocks $c_{b_1}, ..., c_{b_n}$, and an

adequate way to recover cnt_i. The encryption and decryption operations in this mode are defined as follows.

- Encryption: $c_{b_i} \equiv p_{b_i} \oplus E_k(cnt_i)$;
- Decryption: $p_{b_i} \equiv c_{b_i} \oplus E_k(cnt_i)$.

A.1.2 Symmetric Message Authentication Algorithms

Message authentication algorithms ensure message integrity and the origin authentication of the message. While unkeyed techniques, such as cryptographic hash functions, are used to provide message integrity, symmetric-key techniques, such as Message Authentication Code (*MACs*), also called keyed hash function, are widely used to provide message authentication.

Cryptographic Hash Functions (unkeyed technique). Hash functions are in general compressing methods. They map bit strings of arbitrary finite length to bit strings of fixed length (e.g., n bits with $n \in \mathbb{N}$). That is, a hash function H is defined as $H : \{0,1\}^* \rightarrow \{0,1\}^n$, and its output is called a message digest. Since the message space of H's input is larger than that of its output, collisions are unavoidable. A cryptographic hash function aims at making finding such collisions computationally infeasible (i.e., no method is significantly more efficient than brute-force). More specifically, a cryptographic hash function is characterized by the following security properties:

- Preimage resistance (one-way property): Given a hash output o, it is computationally infeasible to find any input i such that $H(i) = o$.
- Second preimage resistance (weak collision resistance): Given an input i and the corresponding hash output $H(i) = o$, it is computationally infeasible to find a second input $i^{'}$ such that $H(i^{'}) = H(i) = o$.
- Collision resistance: It is computationally infeasible to find any distinct input pairs $i \neq i^{'}$ such that $H(i^{'}) = H(i)$.

An example of a cryptographic hash function that has been widely used in practice is the Secure Hash Algorithm 1 (SHA-1) function. For instance, the derivation of the master key in WLAN from a pre-shared password is based on SHA-1. This function generates an output of 160 bit length from an input of maximum length of 2^{160} bit. In 2005, theoretical attacks on SHA-1 were discovered [Wan05]. This drove the NIST to initiate a public competition to standardize a new cryptographic algorithm, called SHA3. In 2013, the SHA-3 standard was published.

Message Authentication Codes. A message authentication code function, MAC, is a keyed cryptographic hash function. It achieves message integrity and origin authentication based on a symmetric secret key k, i.e., $MAC_k(i) = o$. Only those who have knowledge of the key can successfully generate and verify the MAC digest. A MAC function has the security properties of a cryptographic hash function. In addition, a MAC function has to fulfill the key non-recovery property. That is, it needs to preclude the recovery of the secret key: it should be computationally infeasible to reveal the key, even given that the attacker has several MAC input–output pairs that used that key.
A common approach to realize a MAC function is to use a cryptographic hash function with two inputs, the message and the key. A well-known MAC function that implements this approach is $HMAC$. It comprises an inner and outer hash, and it is defined as follows:

$$HMAC(i) := H(k||padding_1||H(k||padding_2||i))$$

$HMAC$ based on the cryptographic hash function SHA-1 is widely deployed. For instance, it is used in WLAN for key derivations, it is also implemented in both the IPsec protocol suite and the TLS protocol.

A.2 Public-Key Cryptographic Algorithms

The rise of public key cryptography started in the late 1970s based on the seminal papers in [Dif76] and [Mer78]. Public key cryptography complements symmetric-key cryptography to solve the key distribution problem and to achieve the security goal of non-repudiation. In contrast to symmetric-key cryptography, encryption and decryption are not similar operations, and they are not based on the same key (or several keys easily derived from each other). In public-key cryptography, a pair of keys is used, and it is computationally infeasible to compute one key based on the knowledge of the other key. Thus, revealing one of the keys does not compromise the security of the system. Typically, one key is kept private (e.g., the decryption key) while the other is public (e.g., the encryption key). That is why it is called public-key cryptography or asymmetric-key cryptography.
Public-key algorithms are computationally intensive, i.e., they are mainly based on exponentiation operations in Galois fields, and they employ relatively long key sizes (exponents). One of the most established public key algorithms is the RSA algorithm [Riv78]. It was invented by Rivest, Shamir and Adleman in 1977 and was patented in the USA until 2000 [Adl83]. RSA can be used for ciphering and digital signatures.

A.2.1 RSA Ciphering

The RSA ciphering algorithm is a block cipher in which the plaintext and the ciphertext are integers between 0 and $n-1$, where $n \in \mathbb{N}$ must have a size of at least 3072 bit to achieve long term security [Paa09]. The public key in RSA is $K_{pub} := (n, e)$, and the private key is $K_{pr} := (p, q, d)$. Let *mod* denote the modulo operation, $\mathbb{Z}_{\phi(n)}$ the integer ring from 0 to $\phi(n)-1$, and *gcd* the greatest common divisor. The RSA keys are generated as follows:

1. Choose two sufficiently large distinct primes p and q (having approximately the same size);
2. Compute $n = p \cdot q$;
3. Compute $\phi(n) = (p-1) \cdot (q-1)$;
4. Choose a random value e with $0 < e < \phi(n)$, with $gcd(e, \phi(n)) = 1$ $\rightarrow$ e having an inverse in $\mathbb{Z}_{\phi(n)}$;
5. Compute d, the inverse of e: $d \equiv e^{-1} mod\ \phi(n)$.

The encryption and decryption operations in RSA are defined as follows, with p_{b_i} and c_{b_i} being the ith block of the plaintext p and the ciphertext c, respectively.

- Encryption: $c_{b_i} \equiv p_{b_i}^{e} mod\ n$;
- Decryption: $p_{b_i} \equiv c_{b_i}^{d} mod\ n \equiv (p_{b_i}^{e})^{d} mod\ n \equiv p_{b_i}^{ed\ mod \phi(n)} mod\ n \equiv p_{b_i}$.

The security of RSA ciphering is based on the factoring problem, i.e., *finding the prime factors of a positive integer n*. The complexity of this problem is believed to be hard. The problem of revealing d from (e, n) is computationally equivalent to the problem of factoring n. The problem of computing p_{b_i} from c_{b_i} is believed to be computationally equivalent to factoring n.
RSA ciphering is typically used for the secure transport of symmetric keys [Paa09].

A.2.2 RSA Digital Signature

Digital signatures provide message authentication and non-repudiation of the message origin. If a dispute arises regarding the originator of the signed message, a third party can verify the identity of the originator without requiring access to the originator's secret information. Digital signatures are usually appended to the message; they follow a hash and sign paradigm to optimize their performance. First a cryptographic hash function is applied to the message, and then the fixed-size digest value is signed. Only the owner of the private key is able to generate the signature. The RSA digital signature operations are similar to RSA

ciphering, with signing resembling decryption. Let m denote the message and sig the signature. The RSA signature and verification operations are defined as follows.

- Sign: $sig \equiv H(m)^d mod\ n$;
- Verify: $H(m) \equiv sig^e mod\ n \equiv (H(m)^e)^d mod\ n \equiv H(m)^{ed\ mod \phi(n)} mod\ n \equiv H(m)$.

Typically, for the sake of efficiency, a small public key exponent is chosen, in most cases $e \in \{3, 17, 65537\}$, while d, the private key exponent, has similar size to n. As a result, RSA encryption and verification operations are relatively fast, while decryption and signature are computationally expensive.

A.2.3 Asymmetric Key Distribution

When using the RSA cipher or any other public-key ciphering scheme, the public keys need to be authenticated, otherwise, an attacker might impersonate the public key of the intended sender or receiver. A common approach to achieve this goal is to use certificates supported by a trusted certification authority (i.e., a third party). This approach was introduced in 1978 in [Koh78]. A certificate binds the public key of a party with the identity of that party, and it includes other attributes such as a validity time period and a digital signature of the Certification Authority (CA), or a certification path rooted in the trusted CA. That is, the basic form of a certificate *cert* is defined as follows.

$$cert := (k_{pub}, ID, Validity)_{sig_{k_{priv_{CA}}}}$$

Certificates are based on the X.509 ITU standard, they are static, i.e., they need to be re-issued if changes are required. The set of equipment, involved parties, and procedures needed to manage certificates is called the PKI. Using certificates, nodes can securely distribute their public key over insecure channels. Afterwards, the nodes can run a key transport protocol or a key agreement protocol to establish secret symmetric session keys. In the former case, one party generates the secret key and uses, e.g., RSA encryption to send this key encrypted to the other party. In the latter case, the key is derived based on the public and private keys of both parties as well as some exchanged information by running, e.g., the Diffie–Hellman Key Exchange protocol [Paa09]. In this case, both parties control the key value.

B

Brief Introduction to OMNeT++

OMNeT++ stands for 'Objective Modular Network Testbed in C++'. It is a general-purpose tool for discrete event simulations, which is used to simulate a wide range of applications, including communication networks, queuing systems, and business processes [Var08]. In recent years, it has become a very popular network simulator in academia [Kok08, AQ08], and its commercial version, OMNEST, is used by reputable companies such as Cisco, QUALCOMM, and Alcatel-Lucent [OMN].
The implementation of simulation models in OMNeT++ follows a hierarchical architecture. Figure B.1 gives a top-down overview of this architecture. It is based on nested modules of the following three types:

1. *Simple module:* At the lowest level of the module hierarchy, *simple modules* are defined using an easy-to-learn text-based NEtwork Descriptor language (NED) and a C++ implementation. The NED file contains the description of the module in terms of parameters and gates. It offers a friendly user interface allowing of editing the module without needing to recompile the model. The NED file of a *simple module* is always associated with a C++ file that initializes the module during runtime based on the NED parameters. C++ files implement the functionality and the messages of the module. Examples of a simple module are a UDP application or a routing protocol.

2. *Compound module:* At the second level of the module hierarchy, *compound modules* are put together from *simple modules*. The depth of the module nesting is not limited. That is, *compound modules* might be also combined to build a more complex *compound module*. The modules communicate with each other by exchanging messages over their connections (also known as channels). To construct *a compound module*, and to define its configuration parameters and its connections, only the NED language is used. Examples of a *compound module* are wired or wireless routers which include, among others, a routing protocol module, a routing table module, and an interface table module.

3. *System module:* To build the network, *system module*, at the top level of the module hierarchy, the appropriate NED files of the corresponding

submodules are imported into a larger NED file, and the parameters for all the submodules are set appropriately.

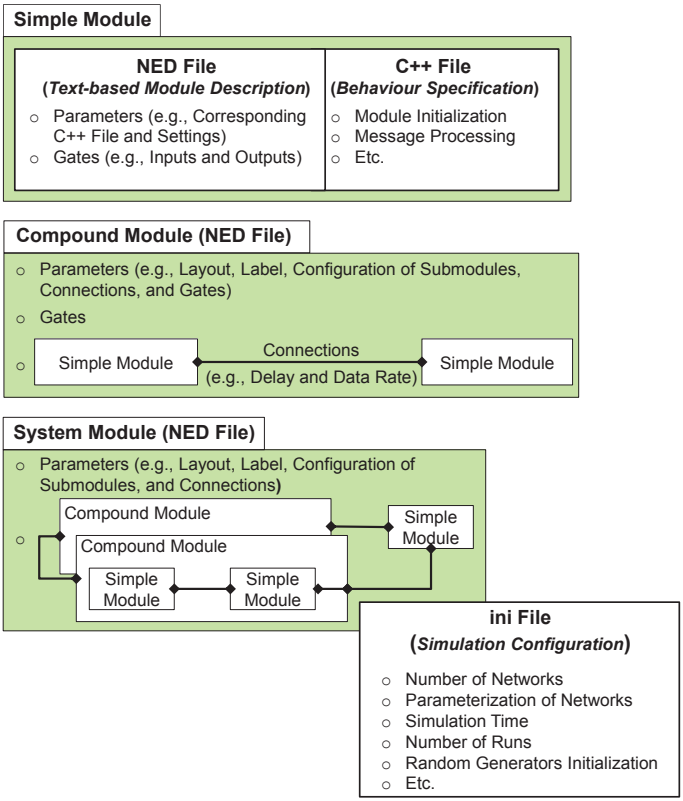

Figure B.1: Overview of OMNeT++ hierarchical architecture.

Upon defining the network, an *ini* file containing general settings is used to execute the simulation.

C

Overview of Modeling WLAN Mesh Networks in INETMANET

An insight on the implementation of WMNs in INETMANET is given in Figure C.1. These are typically modeled as follows.

- The network is composed of several nodes, a channel control module, and a configurator module, see Figure C.1 (top). The nodes represent mobile WLAN mesh backbone nodes. The channel control module is used to handle wireless channel characteristics, e.g., received signal power, interference, and fading. To this end, it always traces the positions of the nodes. The configurator is used to configure the IP and MAC addresses of the nodes.

- The nodes are composed of several modules that implement the relevant components of the TCP/IP stack, see Figure C.1 (bottom-right).

  – At the link layer, several interfaces are defined that model, among others, IEEE 802.11a, IEEE 802.11b, IEEE 802.11g, IEEE 802.11s, and Ethernet. Each interface consists of at least three submodules: a PHY module, a MAC module, and a management module, see Figure C.1 (bottom-right).
  MAC-based routing (e.g., HWMP) is typically simulated as part of the management module, see [AQ09]. Channel models (e.g., air-to-air link of UAVs) are integrated in the PHY module.

  – At the Internet layer, the node implements the IP stack, which includes the IP protocol, ICMP, IGMP, ARP, and others. Besides, a routing manager module is implemented at this layer.
  IP-based routing protocols (e.g., OLSR) are typically simulated as part of the routing manger module, see [AQ08].

  – At the transport layer, INETMANET provides mature implementations of TCP, UDP, SCTP, and other protocols. At the application layer, various traffic types exist, such as HTTP, VoIP, and CBR applications.

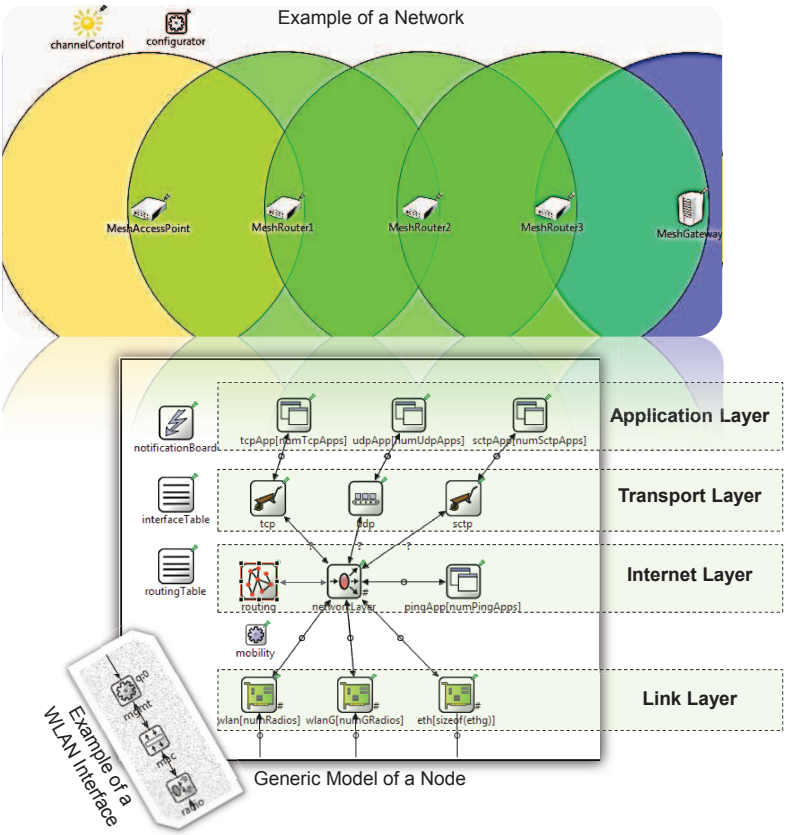

Figure C.1: Generic model of a wireless mesh backbone node in INETMANET.

– Apart from the modules that implement the protocol logic of TCP/IP layers, the node has a routing table, an interface table, a mobility module, and a notification board. The routing and interface tables have a similar structure and functionality to that of Unix-based operating systems. The mobility module can be used to implement specific mobility patterns (e.g., mobility of UAV swarms). The notification board is used to exchange cross-layer notifications between different modules. For instance, the MAC module of the WLAN interface at the link layer uses notifications to report a transmission error to the routing protocol.

D

Scientific Activity Report

Contents of this Chapter

The following scientific contributions are made during this thesis.

D.1 Publications

D.1.1 Journal Submission

1. M. Sbeiti, N. Goddemeier, D. Behnke and C. Wietfeld, *PASER: Position-Aware, Secure, and Efficient Routing Approach for Airborne Mesh Networks*, IEEE Transactions on Wireless Communications - submitted in July 2014.

D.1.2 Conferences

1. M. Sbeiti and C. Wietfeld, *One Stone Two Birds: On the Security and Routing in Wireless Mesh Networks*, IEEE Wireless Communications and Networking Conference (WCNC), Istanbul, Turkey, April 2014.

2. M. Sbeiti and C. Wietfeld, *The Agony of Choice: Behavior Analysis of Routing Protocols in Chain Mesh Networks*, International Conference on Ad Hoc Networks (ADHOCNETS), Barcelona, Spain, October 2013.

3. M. Sbeiti, C. Vogel, A. Wolff and C. Wietfeld, *ROUTE-O-MATIC: A Comprehensive Framework for Reactive Mesh Routing Protocols*, International Conference on Computing, Networking and Communications (ICNC), San Diego, USA, January 2013.

4. M. Sbeiti, J. Hinker and C. Wietfeld, *VLX: A Novel Virtual Localization Extension for Geographical Leash-based Secure Routing in Indoor Wireless Mesh Scenarios*, IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Barcelona, Spain, October 2012.

5. M. Sbeiti, J. Pojda and C. Wietfeld, *Performance Evaluation of PASER - an Efficient Secure Route Discovery Approach for Wireless Mesh Networks*, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Sydney, Australia, September 2012.

6. A. Wolff, M. Sbeiti and C. Wietfeld, *Performance Evaluation of Process-Oriented Wireless Relay Deployment in Emergency Scenarios*, IEEE Symposium on Computers and Communications (ISCC), Cappadocia, Turkey, July 2012.

7. J. Somorovsky, C. Meyer, T. Tran, M. Sbeiti, J. Schwenk and C. Wietfeld, *Sec2: Secure Mobile Solution for Distributed Public Cloud Storages*, International Conference on Cloud Computing and Services Science (CLOSER), Porto, Portugal, April 2012.

8. M. Sbeiti, T. Tran, S. Subik, A. Wolff and C. Wietfeld, *MuSE: Novel Efficient Multi-Tier Communication Security Model for Emergency and Rescue Operations*, IEEE MASS Workshop on Mobile Ad-Hoc Networks for Public Safety Systems (WMAPS), Valencia, Spain, October 2011.

9. J. Pojda, A. Wolff, M. Sbeiti and C. Wietfeld, *Performance Analysis of Mesh Routing Protocols for UAV Swarming Applications*, International Symposium on Wireless Communication Systems (ISWCS), Aachen, Germany, November 2011.

10. M. Sbeiti, A. Wolff and C. Wietfeld, *PASER: Position Aware Secure and Efficient Route Discovery for Wireless Mesh Networks*, International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), Nice, France, August 2011.

11. T. Tran, M. Sbeiti and C. Wietfeld, *A novel Role- and Certificate-based Single Sign-On System for Emergency Rescue Operations*, IEEE International Conference on Communications (ICC), Kyoto, Japan, June 2011.

12. S. Traboulsi, M. Sbeiti, D. Szczesny, A. Showk and A. Bilgic, *High-Performance and Energy-Efficient Sliced AES Multi-Block Encryption for LTE Mobile Devices*, International Conference on Computer and Communication Devices (ICCCD), Bali Island, Indonesia, April 2011.

13. S. Traboulsi, M. Sbeiti, F. Bruns, S. Hessel and A. Bilgic, *An Optimized Parallel and Energy-Efficient Implementation of SNOW 3G for LTE Mobile Devices*, IEEE International Conference on Communication Technology (ICCT), Nanjing, China, November 2010.

14. M. Sbeiti, M. Silbermann, A. Poschmann and C. Paar, *Design Space Exploration of Present Implementations for FPGAs*, Southern Programmable Logic Conference (SPL), Sao Carlos, Brazil, April 2009.

D.1.3 Poster & Code Contribution

1. M. Sbeiti and C. Wietfeld, *On the Implementation Code of the Secure Mesh Routing Protocol PASER in OMNeT++: The Big Picture*, International Workshop on OMNeT++, Cannes, France, March 2013.

D.2 Patent Application

1. C. Wietfeld, M. Sbeiti, S. and A. Wolff, *Access Method and Communication System for Accessing a Protected Communication Service*, WO/2013/056737, April 2013.

D.3 Internet Draft

1. M. Sbeiti and C. Wietfeld, *PASER: Position Aware Secure and Efficient Mesh Routing Protocol*, IETF Internet Draft, draft-sbeiti-karp-paser-00, November 2012.

D.4 Scientific Activities

D.4.1 Technical Program Committee Member

- IARIA SECURWARE 2014, 2013, 2012
  (International Conference on Emerging Security Information, Systems and Technologies)
- SAI 2014, 2013
  (Science and Information Conference)

D.4.2 Session Chair

- IEEE WCNC 2014
  (Wireless Communications and Networking Conference)
- ADHOCNETS 2013
  (International Conference on Ad Hoc Networks)

D.4.3 Reviewer

- IEEE Transactions on Wireless Communications 2014, 2013
- IEEE Transactions on Smart Grid 2014
- International Journal of Distributed Sensor Networks 2014
- IEEE APWiMob 2014
  (Asia Pacific Conference on Wireless and Mobile Technologies)
- SNDS 2014
  (International Conference on Security in Computer Networks and Distributed Systems)
- IEEE GLOBECOM 2013
  (Global Communications Conference)
- IEEE WCNC 2013
  (Wireless Communications and Networking Conference)
- IEEE PIMRC 2012
  (International Symposium on Personal, Indoor and Mobile Radio Communications)
- Wiley European Transactions on Telecommunications 2011, 2010

D.5 Contributions to Collaborative Research Projects

1. Contributing to the IT Security and the Demonstrator of the European FP7 Research Project **SecInCoRe** *(Secure Dynamic Cloud for Information, Communication and Resource Interoperability based on Pan-European Disaster Inventory).*

2. Responsible for the German MIWF-NRW Project **HISEC-GKM** Dealing with a Prototype Implementation and Exploitation of a Patent Application *(Highly-secure Efficient Out-of-Band Over-the-Air Group Key Management in Public Safety Networks).*

3. Responsible for the Evaluation of Aerial Mesh Networks in the European FP7 Research Project **AIRBEAM** (*AIRborne information for Emergency situation Awareness and Monitoring).*

4. Responsible for the Ad Hoc Networks Area in the European/NRW Goal2 Research Project **AVIGLE** (*Avionic Digital Service Platform).*

5. Contributing to the IT Security and Handover Solutions in the the German BMBF Research Project **Sec**^{2} (*Secure Ad-hoc On Demand Virtual Private Storage).*

6. Contributions for the Specification and Implementation of the Protection and Rescue Markup Language (PRML); Technical Support of Interdisciplinary Partners in the German BMBF Research Project **SPIDER** (*Security System for Public Institutions in Disastrous Emergency ScenaRios).*

D.6 Supervision of Student Theses

1. Majuran Rajakanthan, *Leistungsbewertung von Routingprotokollen und Sicherheitslösungen für Wireless Mesh Netze anhand diverser Szenarien in OMNeT++*, Bachelor Thesis, June 2014.

2. Jan Schröder, *Theoretical and Simulation-Based Performance Analysis of Secure Mesh Routing Protocols in OMNeT++ and ns-3*, Bachelor Thesis, June 2014.

3. Mohamad Nehme, *Leistungsbewertung und Sicherheitsanalyse von HWMP/IEEE802.11s in OMNeT++*, Study Thesis, August 2013.

4. Jonas Hinker, *Erweiterung des PASER-Protokolls zur Sicherung von Knoten ohne GPS-Empfang vor Wormhole-Angriffe*, Bachelor Thesis, July 2012.

5. Carsten Vogel, *Experimental Implementation of a Comprehensive Framework for PASER*, Study Thesis, June 2012.
6. Eugen Paul, *Simulative Leistungsbewertung des WMN-Routingprotokolls PASER*, Master Thesis, May 2012.
7. Daniel Merget, *Improving the IKEv2 Protocol Robustness in Error-Prone Channels*, Bachelor Thesis, November 2011 —Joint work with Rohde & Schwarz.
8. Lorenzo Benet Gonzalez, *Leistungsbewertung des AODV-Routingprotokolls in einem Wireless-Mesh-Netz während eines Wormhole-Angriffs in OMNeT++*, Study Thesis, September 2011.
9. Giuseppe Tabbi, *Konzept und Analyse eines rollen- und zertifikatbasierten Single Sign-On*, Study Thesis, September 2010 —Joint work with Dr.-Ing. Thang Tran.

D.7 Mentoring of Seminars

1. Wen Cui, Marcus Hafercamp, Sebastian Schellhase and Benjamin Sliwa, *Enabling Next Generation Airborne Communications*, September 2014 — Joint work with Dipl.-Inf. Daniel Behnke and Dipl.-Inf. Maike Kuhnert.
2. Thierry Itondo, Lionel Cedric Noukahoua, Oussama Renuli, Daniel Swientek and Carsten Vogel, *Security and Performance of Wireless Mesh Networks*, June 2011.
3. Huy Cao Tri Do, Claudia Gross, Christian Kay, Gia Vinh Luong, Giuseppe Tabbi and Anton Tripolez, *Security of Routing Protocols in Wireless Mesh Networks*, December 2010 —Joint work with Dr.-Ing. Thang Tran.

D.8 Teaching

- Co-supervision of a practical course dealing with simulation of communication networks in winter term 2013/2014
- Support of a lecture on networked mobile robot systems in winter term 2013/2014
- Support of a lecture on model-based dimensioning of communication systems in winter term 2013/2014
- Support of a lecture on communication networks in winter term 2012/2013

List of Acronyms

MACs Message Authentication Code.

AARF Adaptive Auto Rate Fallback.

AES Advanced Encryption Standard.

AMPE Authenticated Mesh Peering Exchange protocol.

ANUM Authentication Neighbor Update Message.

AoA Angle of Arrival.

AODV Ad Hoc On Demand Distance Vector protocol.

API Application Programming Interface.

APIT Approximate Point In Triangle Test.

ARAN Authenticated Routing for Ad-hoc Networks.

ARF Auto Rate Fallback.

ASL Ad-hoc Support Library.

BATMAN Better Approach To Mobile Ad-hoc Networking.

CA Certification Authority.

CAPF Communication-Aware Potential Fields.

CASTOR Continuously Adapting Secure Topology-Oblivious Routing.

CBR Constant Bit Rate.

CCMP Cipher block Chaining Message authentication code Protocol.

CNI Communication Networks Institute.

CNPC Command and Non-Payload Communication.

CoO Cell of Origin.

CRL Certificate Revocation List.

CSMA/CA Carrier Sense Multiple Access/Collision Avoidance.

CTK Clien Transient Key.

CTR CounTeR mode.

CTS Clear-To-Send.

DBPSK Differential Binary Phase Shift Keying.

DCF Distributed Coordination Function.

DelPHI Delay Per Hop Indication.

DSSS Direct Sequence Spread Spectrum.

ECC Elliptic Curve Cryptography.

ECDSA Elliptic Curve Digital Signature Algorithm.

EDCA Enhanced Distributed Channel Access.

ETX Expected Transmission Count.

GTK Group Network Key.

HWMP Hybrid Wireless Mesh Protocol.

IBC Identity Based Cryptography.

IEEE Institute of Electrical and Electronics Engineers.

IETF Internet Engineering Task Force.

IPsec Internet Protocol security.

KDC Key Distribution Center.

LLF Link Layer Feedback.

LQ Link Quality.

LTE Long Term Evolution.

LTE-TDD LTE-Time Division Duplex.

MAC Medium Access Control.

MANETs Mobile Ad-hoc NETworks.

MCCA Mesh Coordination Controlled Channel Access.

MIMO Multiple Input Multiple Output.

MPRs MultiPoint Relays.

NED NEtwork Descriptor language.

NHPS NeighborHood Discovery Protocol.

NIST National Institute of Standards and Technology.

NLQ Neighbour Link Quality.

NSA National Security Agency.

OGMs OriGinator messages.

OLSR Optimized Link State Routing.

PASER Position-Aware, Secure, and Efficient mesh Routing.

PDF Probability Density Function.

PDR Packet Delivery Ratio.

PKI Public key Infrastructure.

PREP Path RESponse.

PREQ Path REQuest mechanism.

PRNG Pseudo Random Number Generator.

PSI Position Swap Interval.

QoS Quality Of Service.

RANN Root ANNouncement.

RAOLSR Radio Aware OLSR.

RC4 Rivest Cipher 4.

RERR Route ERRor.

RM Radio Metric.

RREP Route REPly.

RREQ Route REQuest.

RSN Robust Secure Network.

RSS Received Signal Strength.

RTS Request-To-Send.

RTT Round-Trip Time.

SAE Simultaneous Authentication of Equals.

SAODV Secure Ad-hoc On-demand Distance Vector.

SEAD Secure Efficient Ad-hoc Distance vector.

SEAODV Security Enhanced AODV.

SECTOR SECure Tracking Of node encounteRs.

SHA Secure Hash Algorithm.

SHWMP Secure HWMP.

SNIR Signal to Noise plus Interference Ratio.

SNR Signal-to-Noise Ratio.

SOLSR Secure Optimized Link State Routing.

SWMP Secure Wireless Mesh Protocol.

TC Topology Control.

TDoA Time Difference of Arrival.

TETRA Terrestrial Trunked Radio.

TKIP Temporal Key Integrity Protocol.

TLS Transport Layer Security.

ToA Time of Arrival.

UAVs Unmanned Aerial Vehicles.

VLX Virtual Localization Extension.

WAP Wormhole Attack Prevention.

WCA Weighted Centroid Algorithm.

WEP Wired Equivalent Privacy.

WLAN Wireless Local Area Network.

WMN WLAN Mesh Network.

WPA Wi-Fi Protected Access.

WPA2 Wi-Fi Protected Access 2.

References

[Abd12] A. Abdulla et al. *HYMN: A Novel Hybrid Multi-Hop Routing Algorithm to Improve the Longevity of WSNs. IEEE Transactions on Wireless Communications*, volume 11(7), 2012.

[Abo03] B. Aboba. *Fast Handoff Issues.* In *IEEE 802.11-03/155r0.* Mar. 2003. [Online]. Available: `https://mentor.ieee.org/802.11/dcn/03/11-03-0155-00-000i-fast-handoff-issues.ppt`.

[Abo04] M. Abolhasan, T. Wysocki and E. Dutkiewicz. *A Review of Routing Protocols for Mobile Ad Hoc Networks. Elsevier Ad Hoc Networks*, volume 2(1), 2004.

[Abo09] M. Abolhasan, B. Hagelstein and J. Wang. *Real-World Performance of Current Proactive Multi-Hop Mesh Protocols.* In *IEEE Asia-Pacific Conference on Communications (APCC).* Oct. 2009.

[Abu08] L. Abusalah, A. Khokhar and M. Guizani. *A Survey of Secure Mobile Ad hoc Routing Protocols. IEEE Communications Surveys and Tutorials*, volume 10(4), 2008.

[Adj03] C. Adjih, E. Baccelli and P. Jacquet. *Link State Routing in Wireless Ad-hoc Networks.* In *IEEE Military Communications Conference (MILCOM).* Oct. 2003.

[Adl83] L. M. Adleman, R. L. Rivest and A. Shamir. *Cryptographic Communications System and Method. US Patent 4,405,829.*, Sep. 1983.

[AIR] *AIRBorne information for Emergency situation Awareness and Monitoring (AIRBEAM).* European Research Project (FP7). [Online]. Available: `http://airbeam.eu/project/`. Accessed: 2014-09-01.

[Aky05] I. Akyildiz, X. Wang and W. Wang. *Wireless Mesh Networks: A Survey. Elsevier Computer Networks*, volume 47(4), 2005.

[ANC] *UAV-Assisted Ad Hoc Networks for Crisis Management and Hostile Environment Sensing (ANCHORS).* German-French Research Project (BMBF/ANR). [Online]. Available: `http://anchors-project.org/index.php/en/`. Accessed: 2014-09-01.

[And06] T. Andel and A. Yasinsac. *On the Credibility of Manet Simulations. IEEE Computer*, volume 39(7), 2006.

[And07] T. Andel and A. Yasinsac. *Surveying Security Analysis Techniques in MANET Routing Protocols. IEEE Communications Surveys Tutorials*, volume 9(4), 2007.

[Aok06] H. Aoki, S. Takeda, K. Yagyu and A. Yamada. *IEEE 802.11s Wireless LAN Mesh Network Technology. NTT DoCoMo Technical Journal*, volume 8(2), 2006.

[AQ08] A. Ariza-Quintana, E. Casilari and A. T. n. Cabrera. *Implementation of MANET Routing Protocols on OMNeT++.* In *ICST International Conference on Simulation Tools and Techniques (SIMUTools)*. Mar. 2008.

[AQ09] A. Ariza-Quintana, E. Casilari and A. T. n. Cabrera. *An Architecture for the Implementation of Mesh Networks in OMNeT++.* In *ICST International Conference on Simulation Tools and Techniques (SIMUTools)*. Mar. 2009.

[AR12] M. Al-Rabayah and R. Malaney. *A New Scalable Hybrid Routing Protocol for VANETs. IEEE Transactions on Vehicular Technology*, volume 61(6), 2012.

[Asa13] M. Asadpour, D. Giustiniano, K. A. Hummel, S. Heimlicher and S. Egli. *Now or Later?: Delaying Data Transfer in Time-critical Aerial Communication.* In *ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT)*. Dec. 2013.

[Aus10] R. Austin. *Unmanned Aircraft Systems: UAVS Design, Development and Deployment.* Wiley, 2010.

[AVI] *Avionic Digital Service Platform (AVIGLE).* German Research Project (EU/NRW-Ziel2). [Online]. Available: `http://www.avigle.de/main/index.php?lang=en`. Accessed: 2014-09-01.

[AWM] *Athens Wireless Metropolitan Network (AWMN).* [Online]. Available: `http://www.athenswireless.net/`. Accessed: 2014-09-01.

[Bah07] M. Bahr. *Update on the Hybrid Wireless Mesh Protocol of IEEE 802.11s.* In *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems (MASS)*. Oct. 2007.

[Bak03] F. Baker, M. Chandra, R. White, J. Macker, T. Henderson and E. Baccelli. *Problem Statement for OSPF Extensions for Mobile Ad Hoc Routing.* Internet-Draft: https://tools.ietf.org/id/draft-baker-manet-ospf-problem-statement-00.txt. Stream: IETF Mobile Ad Hoc and OSPF Working Groups, Sep. 2003.

[Bal14] G. Baldini, S. Karanasios, D. Allen and F. Vergari. *Survey of Wireless Communication Technologies for Public Safety. IEEE Communications Surveys & Tutorials*, volume 16(2), 2014.

[Bar09] L. Barolli, M. Ikeda, G. DeMarco, A. Durresi and F. Xhafa. *Performance Analysis of OLSR and BATMAN Protocols Considering Link Quality Parameter.* In *IEEE International Conference on Advanced Information Networking and Applications (AINA)*. May 2009.

[BATa] *Better Approach To Mobile Adhoc Networking (B.A.T.M.A.N.).* Freifunk Community. [Online]. Available: `http://open-mesh.net/batman/`. Accessed: 2014-09-01.

[BATb] *Branches of the BATMAN Routing Protocol.* [Online]. Available: `http://www.open-mesh.org/projects/open-mesh/wiki/BranchesExplained`. Accessed: 2014-09-01.

[BATc] *Wireless Battle Mesh.* [Online]. Available: `http://battlemesh.org/AboutUs`. Accessed: 2014-09-01.

[Beh13] D. Behnke, P.-B. Bök and C. Wietfeld. *UAV-Based Connectivity Maintenance for Borderline Detection.* In *IEEE Vehicular Technology Conference (VTC)*. Sep. 2013.

[Ber14a] D. J. Bernstein and T. Lange. *Batch NFS.* In *Selected Areas in Cryptography (SAC)*. Aug. 2014.

[Ber14b] D. J. Bernstein, T. Lange and R. Niederhagen. *The Basic Back Door to Dual Elliptic Curves.* [Online]. Available: `https://projectbullrun.org/dual-ec/back-door.html`, Mar. 2014. Accessed: 2014-09-01.

[Bia98] G. Bianchi. *IEEE 802.11 - Saturation Throughput Analysis. IEEE Communications Letters*, volume 2(12), 1998.

[Bic05a] J. Bicket, D. Aguayo, S. Biswas and R. Morris. *Architecture and Evaluation of an Unplanned 802.11b Mesh Network.* In *ACM International Conference on Mobile Computing and Networking (MobiCom)*. Sep. 2005.

[Bic05b] J. C. Bicket. *Bit-Rate Selection in Wireless Networks.* Technical report, Master Thesis, Massachusetts Institute of Technology (MIT), USA, Feb. 2005.

[Bir09] T. Bird. *Measuring function duration with ftrace.* In *Linux Symposium*. Oct. 2009.

[Bök14] P.-B. Bök, K. Kohls, D. Behnke and C. Wietfeld. *Distributed Flow Permission Inspection for Mission-Critical Communication of Untrusted Autonomous Vehicles*. In *IEEE Vehicular Technology Conference (VTC)*. Sep. 2014.

[Blu07] J. Blumenthal, R. Grossmann, F. Golatowski and D. Timmermann. *Weighted Centroid Localization in Zigbee-based Sensor Networks*. In *IEEE International Symposium on Intelligent Signal Processing (WISP)*. Oct. 2007.

[BO13] J. Ben-Othman and Y. Saavedra Benitez. *IBC-HWMP: a Novel Secure Identity-based Cryptography-based Scheme for Hybrid Wireless Mesh Protocol for IEEE 802.11s. Wiley Online Library on Concurrency and Computation: Practice and Experience*, volume 25(5), 2013.

[Bob03] R. Bobba, L. Eschenauer, V. Gligor and W. Arbaugh. *Bootstrapping Security Associations for Routing in Mobile Ad-hoc Networks*. In *IEEE Global Communications Conference (GLOBECOM)*. Dec. 2003.

[Bre09] M. Bredel and M. Bergner. *On the Accuracy of IEEE 802.11g Wireless LAN Simulations Using OMNet++*. In *International Workshop on OMNeT++*. Mar. 2009.

[Büs14] M. Büscher, M. Kuhnert, M. Ahlsen, J. Pottebaum, B. VanVeelen, C. Easton and C. Wietfeld. *Cloud Computing for Disaster Response: Ethical, Legal, Social Issues*. In *International Conference on Information Systems for Crisis Response and Management (ISCRAM)*. May 2014.

[BUG13a] *Bug Fix in in the Final State Machine of IEEE 802.11 MAC in INETMANET*. Online. [Available]: `https://groups.google.com/forum/`!msg/omnetpp/UGsTNVQ5SuE/YbF-SSVkbPwJ, Mar. 2013. Accessed: 2014-09-01.

[BUG13b] *Bug Fix in in the Frame Error Rate Calculation of IEEE 802.11 Radio in INETMANET*. Online. [Available]: `https://groups.google.com/forum/`!topic/omnetpp/_MsDeUc1vz4, Feb. 2013. Accessed: 2014-09-01.

[Bul00] N. Bulusu, J. Heidemann and D. Estrin. *GPS- Less Low-Cost Outdoor Localization for Very Small Devices. IEEE Personal Communications*, volume 7(5), 2000.

[Bur07] M. Burmester and B. D. Medeiros. *Towards Provable Security for Route Discovery Protocols in Mobile Ad Hoc Networks*. In *IACR Cryptology ePrint Archive*. 2007.

[Bur09] M. Burmester and B. de Medeiros. *On the Security of Route Discovery in MANET. IEEE Transactions on Mobile Computing*, volume 8(9), 2009.

[Cap03] S. Capkun, L. Buttyán and J.-P. Hubaux. *SECTOR: Secure Tracking of Node Encounters in Multi-Hop Wireless Networks.* In *ACM Workshop on Security of Ad-Hoc and Sensor Networks (SASN).* Oct. 2003.

[Car11] R. Carrano, L. Magalhaes, D. Saade and C. Albuquerque. *IEEE 802.11s Multihop MAC: A Tutorial. IEEE Communications Surveys and Tutorials*, volume 13(1), 2011.

[Cha04] P. Chatzimisios, A. Boucouvalas and V. Vitsas. *Effectiveness of RTS/CTS Handshake in IEEE 802.11a Wireless LANs. IET Electronics Letters*, volume 40(14), 2004.

[Cha05a] I. D. Chakeres and E. M. Belding-Royer. *AODV Implementation Design and Performance Evaluation. International Journal of Wireless and Mobile Computing (IJWMC)*, volume 2(3), 2005.

[Cha05b] H. Chan, V. D. Gligor, A. Perrig and G. Muralidharan. *On the Distribution and Revocation of Cryptographic Keys in Sensor Networks. IEEE Transactions on Dependable and Secure Computing*, volume 2(3), 2005.

[Cha12] Z. Chang, O. Alanen, T. Huovinen, T. Nihtila, E. H. Ong, J. Kneckt and T. Ristaniemi. *Performance Analysis of IEEE 802.11ac DCF with Hidden Nodes.* In *IEEE Vehicular Technology Conference (VTC).* Feb. 2012.

[Che05] L. Chen, S. Low and J. Doyle. *Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks.* In *IEEE Conference on Computer Communications (INFOCOM).* Mar. 2005.

[Che06] O. Cheikhrouhou, M. Laurent-Maknavicius and H. Chaouchi. *Security Architecture in a Multi-Hop Mesh Network.* In *Conference on Security and Network Architectures (SAR).* Jun. 2006.

[Chi06] H. S. Chiu and K.-S. Lui. *DelPHI: Wormhole Detection Mechanism for Ad Hoc Wireless Networks.* In *IEEE International Symposium on Wireless and Pervasive Computing (ISWPC).* Jan. 2006.

[Chl12] P. Chlumsky, Z. Kocur and J. Vordazka. *Comparison of Different Scenarios for Path Diversity Packet Wireless Networks.* volume 3(1), 2012.

[Cho08] S. Choi, D. young Kim, D. hyeon Lee and J. il Jung. *WAP: Wormhole Attack Prevention Algorithm in Mobile Ad Hoc Networks*. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*. Jun. 2008.

[Cla90] B. N. Clark, C. J. Colbourn and D. S. Johnson. *Unit Disk Graphs. Discrete Mathematics*, volume 86(3), 1990.

[Cla03] T. Clausen and P. Jacquet. *Optimized Link State Routing Protocol (OLSR)*. RFC 3626. Status: Experimental. Stream: IETF Network Working Group, Oct. 2003.

[Cla11] T. Clausen, C. Dearlove and J. Dean. *Mobile Ad Hoc Network (MANET) Neighborhood Discovery Protocol (NHDP)*. RFC 6130. Status: Standards Track. Stream: IETF, Apr. 2011.

[CTR80] *DES Modes of Operation*. National Institute of Standards and Technology (NIST). [Online]. Available: `http://csrc.nist.gov/publications/fips/fips81/fips81.htm`, Dec. 1980. Accessed: 2014-09-01.

[Cur04] J. Curry, J. Maslanik, G. Holland and J. Pinto. *Applications of Aerosondes in the Arctic. Bulletin of the American Meteorological Society*, volume 85(12), 2004.

[Dan09] K. Daniel, B. Dusza, A. Lewandowski and C. Wietfeld. *AirShield: A System-of-Systems MUAV Remote Sensing Architecture for Disaster Response*. In *IEEE Systems Conference (SysCon)*. Mar. 2009.

[Dan10] K. Daniel, A. Wolff and C. Wietfeld. *Protocol Design and Delay Analysis for a MUAV-Based Aerial Sensor Swarm*. In *IEEE Wireless Communications and Networking Conference (WCNC)*. Apr. 2010.

[DeC05] D. DeCouto, D. Aguayo, J. Bicket and R. Morris. *A High-Throughput Path Metric for Multi-Hop Wireless Routing. Springer Wireless Networks*, volume 11(4), 2005.

[Dif76] W. Diffie and M. Hellman. *New Directions in Cryptography. IEEE Transactions on Information Theory*, volume 22(6), 1976.

[Dur02] G. D. Durgin. *Space-Time Wireless Channels*. Prentice Hall, 2002.

[Dut07] P. Dutta, S. Jaiswal, D. Panigrahi, K. Naidu, R. Rastogi and A. Todimala. *VillageNet: A Low-Cost, 802.11-Based Mesh Network for Rural Regions*. In *ICST International Conference on Communication Systems Software and Middleware (COMSWARE)*. Jan. 2007.

[Edn03] J. Edney and W. Arbaugh. *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Addison-Wesley, 2003.

[Egn10] A. Egners and U. Meyer. *Wireless Mesh Network Security: State of Affairs*. In *IEEE Conference on Local Computer Networks (LCN)*. Oct. 2010.

[Egn12] A. Egners, H. Fabelje and U. Meyer. *FSASD: A Framework for Establishing Security Associations for Sequentially Deployed WMN*. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. Jun. 2012.

[Eme15] *Nach Snowden: Wenig Schlaf für Kryptoforscher*. [Online]. Available: `http://www.heise.de/security/artikel/Nach-Snowden-Wenig-Schlaf-fuer-Kryptoforscher-2392236.html`, 2015. Accessed: 2015-03-18.

[FAA] *New Rules for Small Unmanned Aircraft Systems*. [Online]. Available: `http://www.faa.gov/news/press_releases/news_story.cfm?newsId=18295\&cid=TW299`. Accessed: 2015-06-03.

[FAC] *Facebook Will Deliver Internet Via Drones*. [Online]. Available: `http://techcrunch.com/2014/03/27/facebook-drones/`. Accessed: 2015-06-03.

[FIR] *HotPort 5020 Wireless Mesh Nodes*. firetide. [Online]. Available: `http://www.firetide.com/products/mesh-infrastructure/hotport-5020/`. Accessed: 2014-09-01.

[FRE] *Freifunk Berlin*. [Online]. Available: `http://berlin.freifunk.net/`. Accessed: 2014-09-01.

[Fri46] H. Friis. *A Note on a Simple Transmission Formula. Proceedings of the Institute of Radio Engineers (IRE)*, volume 34(5), 1946.

[Ful97] C. Fullmer and J. Garcia-Luna-Aceves. *Solutions to Hidden Terminal Problems in Wireless Networks. ACM Computer Communication Review*, 1997.

[Gal10] W. Galuba, P. Papadimitratos, M. Poturalski, K. Aberer, Z. Despotovic and W. Kellerer. *Castor: Scalable Secure Routing for Ad Hoc Networks*. In *IEEE International Conference on Computer Communications (INFOCOM)*. Mar. 2010.

[GEN09] *Generic Netlink HOWTO*. Linux Foundation. [Online]. Available: `http://www.linuxfoundation.org/collaborate/workgroups/networking/\\generic\_netlink\_howto`, 2009. Accessed: 2014-09-01.

[God] N. Goddemeier. *Communication and Cooperation Strategies for Aerial Robotic Systems (Ongoing Work)*. Ph.D. thesis, TU Dortmund University, Germany.

[God11] N. Goddemeier, S. Rohde, J. Pojda and C. Wietfeld. *Evaluation of Potential Fields Mobility Strategies for Aerial Network Provisioning*. In *IEEE Global Communications Conference (GLOBECOM). Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (Wi-UAV)*. Dec. 2011.

[God12a] N. Goddemeier, K. Daniel and C. Wietfeld. *Role-Based Connectivity Management with Realistic Air-to-Ground Channels for Cooperative UAVs. IEEE Journal on Selected Areas in Communications*, volume 30(5), Jun. 2012.

[God12b] N. Goddemeier, S. Rohde and C. Wietfeld. *Experimental Validation of RSS Driven UAV Mobility Behaviors in IEEE802.11s Networks*. In *IEEE Global Communications Conference (GLOBECOM). Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (Wi-UAV)*. Dec. 2012.

[God14] N. Goddemeier, S. Rohde and C. Wietfeld. *Experimental Performance Evaluation of Role-Based Connectivity Management for Cooperating UAVs*. In *IEEE Vehicular Technology Conference (VTC)*. Sep. 2014.

[Gol11] S. Gollakota and D. Katabi. *Physical Layer Wireless Security Made Fast and Channel Independent*. In *IEEE International Conference on Computer Communications (INFOCOM)*. April 2011.

[GOO] *Google Wifi in Mountain View*. [Online]. Available: `http://wifi.google.com/city/mv/apmap.html`. Accessed: 2014-09-01.

[GPS] *GPSd API*. [Online]. Available: `http://www.aosabook.org/en/gpsd.html`. Accessed: 2014-12-22.

[Gär97] B. Gärtner and S. Schönherr. *Exact Primitives for Smallest Enclosing Ellipses*. In *ACM Symposium on Computational Geometry (SoCG)*. Jun. 1997.

[Grö13] S. Gröning, C. Rosas and C. Wietfeld. *COMPLeTe: A COMmunication Protocol vaLidation Toolchain*. In *International SPIN Symposium on Model Checking of Software*. Jul. 2013.

[GUI] *Guifi.net: Open, Free and Neutral Network and Internet for Everybody*. Cisco. [Online]. Available: `http://guifi.net/en`. Accessed: 2014-09-01.

[Har08] D. Harkins. *Simultaneous Authentication of Equals: A Secure, Password-Based Key Exchange for Mesh Networks.* In *International Conference on Sensor Technologies and Applications (SENSORCOMM)*. Aug. 2008.

[He03] T. He, C. Huang, B. M. Blum, J. A. Stankovic and T. Abdelzaher. *Range-Free Localization Schemes for Large Scale Sensor Networks.* In *ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*. Sep. 2003.

[Hie08] G. Hiertz, S. Max, T. Junge, D. Denteneer and L. Berlemann. *IEEE 802.11s - Mesh Deterministic Access.* In *European Wireless Conference (EP)*. Jun. 2008.

[Hie10] G. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann and B. Walke. *IEEE 802.11s: The WLAN Mesh Standard. IEEE Wireless Communications*, volume 17(1), 2010.

[Hie11] G. R. Hiertz. *Mesh Networking Using IEEE 802.11 Wireless Technologies.* Ph.D. thesis, RWTH Aachen, Germany, Dec. 2011.

[Hiy13] M. Hiyama, E. Kulla, M. Ikeda, L. Barolli and M. Takizawa. *Investigation of OLSR Behavior for Different Hello Packets Intervals in a MANET Testbed.* In *IEEE International Conference on Advanced Information Networking and Applications (AINA)*. Mar. 2013.

[Hon05] F. Hong, L. Hong and C. Fu. *Secure OLSR.* In *IEEE Advanced Information Networking and Applications (AINA)*. Mar. 2005.

[Hsu04] J. Hsu, S. Bhatia, K. Tang, R. Bagrodia and M. Acriche. *Performance of Mobile Ad hoc Networking Routing Protocols in Large Scale Scenarios.* In *IEEE Military Communications Conference (MILCOM)*. Oct. 2004.

[Hu03a] Y. Hu, D. Johnson and A. Perrig. *SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. Elsevier Ad Hoc Netowrks*, volume 1(1), 2003.

[Hu03b] Y. Hu, A. Perrig and D. Johnson. *Packet Leashes: A Defense Against Wormhole Attacks in Wireless Networks.* In *IEEE International Conference on Computer Communications (INFOCOM)*. Mar. 2003.

[Hu04] Y. Hu and A. Perrig. *A Survey of Secure Wireless Ad hoc Routing. IEEE Security and Privacy*, volume 2(3), 2004.

[Hu05] Y. Hu, A. Perrig and D. Johnson. *Ariadne: A Secure On-Demand Routing Protocol for Ad hoc Networks. ACM Journal on Wireless Networks*, volume 11(1-2), 2005.

[Huh04] A. Huhtonen. *Comparing AODV and OLSR routing protocols.* In *Seminar on Internetworking. Telecommunication Software and Multimedia Laboratory. Helsinki University of Technology, Finland.* Jun. 2004.

[IEE03] *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part Ii: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Sep. 2003.

[IEE04] *IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Medium Access Control (MAC) Security Enhancements.* IEEE Std. 802.11i-2004, Aug. 2004.

[IEE11] *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 10: Mesh Networking*, Sep. 2011.

[IEE13a] *IEEE Standard for Information Technology– Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks– Specific Requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4: Enhancements for Very High Throughput for Operation in Bands below 6 GHz.* IEEE Std 802.11ac-2013, Dec. 2013.

[IEE13b] *IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.* IEEE Std. 802.11-2012, Mar. 2013.

[IETa] *Keying and Authentication for Routing Protocols Group.* IETF. [Online]. Available: `https://tools.ietf.org/wg/karp/charters`. Accessed: 2014-09-01.

[IETb] *Mobile Ad-hoc Networks Group (MANET))*. [Online]. Available: `http://datatracker.ietf.org/wg/manet/charter/`. Accessed: 2014-09-01.

[INEa] *INET Framework*. [Online]. Available: `http://inet.omnetpp.org/`. Accessed: 2014-09-01.

[INEb] *INETMANET Framework*. [Online]. Available: `https://github.com/aarizaq/inetmanet-2.0`. Accessed: 2014-09-01.

[INE13] *PASER integrated in the INETMANET framework*. [Online]. Available: `https://github.com/aarizaq/inetmanet-2.0/tree/inetmanet-2.2/src/networklayer/manetrouting/PASER`, Jul. 2013. Accessed: 2014-12-22.

[Isl09] M. Islam, M. Hamid and C. Hong. *SHWMP: A Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s Wireless Mesh Networks*. In *Springer Transactions on Computational Science VI*, volume 5730. 2009.

[Joh99] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek and M. Degermark. *Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks*. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. Aug. 1999.

[Joh08] D. Johnson, N. Ntlatlapa and C. Aichele. *Simple Pragmatic Approach to Mesh Routing Using BATMAN*. In *IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries (WCITD)*. Oct. 2008.

[Joh10] C. Johnny, J. Wright and V. Liu. *Hacking Exposed Wireless: Wireless Security Secrets & Solutions*. McGraw-Hill, 2010.

[JP12] A. Jimenez-Pacheco, D. Bouhired, Y. Gasser, J. Zufferey, D. Floreano and B. Rimoldi. *Implementation of a Wireless Mesh Network of Ultra Light MAVs with Dynamic Routing*. In *IEEE Global Communications Conference (GLOBECOM). Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (Wi-UAV)*. Dec. 2012.

[JS03] A. K. J. Salim, H. Khosravi and A. Kuznetsov. *Linux Netlink as an IP Services Protocol*. RFC 3549. Status: Infomational. Stream: IETF, Jul. 2003.

[Kah78] R. Kahn, S. Gronemeyer, J. Burchfiel and R. Kunzelman. *Advances in Packet Ratio Technology. Proceedings of the IEEE*, volume 66(11), 1978.

[Kan07] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato and A. Jamalipour. *A Survey of Routing Attacks in Mobile Ad hoc Networks*. *IEEE Wireless Communications*, volume 14(5), 2007.

[Kaw03] V. Kawadia, Y. Zhang and B. Gupta. *System Services for Ad-Hoc Routing: Architecture, Implementation and Experiences*. In *ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*. May 2003.

[Kha05] I. Khalil, S. Bagchi and N. B. Shroff. *LITEWORP: A Lightweight Countermeasure for the Wormhole Attack in Multihop Wireless Networks*. In *IEEE International Conference on Dependable Systems and Networks (DSN)*. Jun. 2005.

[Kha09] M. Khabbazian, H. Mercier and V. K. Bhargava. *Severity Analysis and Countermeasure for the Wormhole Attack in Wireless Ad Hoc Networks*. In *IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*. Sep. 2009.

[Kno13] R. Knobler, P. Scheffel, S. Jackson, K. Gaj and J.-P. Kaps. *Breaking Down the Barriers of Using Strong Authentication and Encryption in Resource Constrained Embedded Systems*. *SPIE Mobile Multimedia/Image Processing, Security, and Applications*, volume 8755, 2013.

[Koh78] L. M. Kohnfelder. *Towards a Practical Public-Key Cryptosystem*. Master's thesis, Massachusetts Institute of Technology (MIT), USA, May 1978.

[Kok08] M. Koksal. *A Survey of Network Simulators Supporting Wireless Networks*. Middle East Technical University, Turkey, Oct. 2008.

[Kuh15] M. Kuhnert, O. Paterour, A. Georgiev, K. Petersen, M. Büscher, J. Pottebaum and C. Wietfeld. *Next Generation, Secure Cloud-based Pan-European Information System for Enhanced Disaster Awareness*. In *International Conference on Information Systems for Crisis Response and Management (ISCRAM)*. May 2015.

[Kun08] A. Kuntz, F. Schmidt-Eisenlohr, O. Graute, H. Hartenstein and M. Zitterbart. *Introducing Probabilistic Radio Propagation Models in OMNeT++ Mobility Framework and Cross Validation Check with ns-3-2*. In *ICST International Conference on Simulation Tools and Techniques (SIMUTools)*. Mar. 2008.

[Kur05] S. Kurkowski, T. Camp and M. Colagrosso. *MANET Simulation Studies: The Incredibles*. In *ACM Mobile Computing and Communications Review*, volume 9. 2005.

[Lac04] M. Lacage, M. H. Manshaei and T. Turletti. *IEEE 802.11 Rate Adaptation: A Practical Approach*. In *ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*. Oct. 2004.

[Leb12] G. Lebovitz and M. Bhatia. *Keying and Authentication for Routing Protocols (KARP) Design Guidelines*. RFC 6518. Status: Infomational. Stream: IETF, Feb. 2012.

[Leb13] G. Lebovitz, M. Bhatia and B. Weis. *Keying and Authentication for Routing Protocols (KARP) Overview, Threats, and Requirements*. RFC 6862. Status: Infomational. Stream: IETF, Mar. 2013.

[Lei06] T. Leinmüller, C. Maihoefer, E. Schoch and F. Kargl. *Improved Security in Geographic Ad Hoc Routing Through Autonomous Position Verification*. In *ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*. Apr. 2006.

[Lew10] A. Lewandowski, V. Koster and C. Wietfeld. *Performance Evaluation of AODV and OLSR-Meshed IP-Enabled IEEE802.15.4*. In *IARIA International Conference on Advances in Mesh Networks (MESH)*. July 2010.

[Lew12] A. Lewandowski. *Wireless Communication for Personal Safety Services: Sensing, Localization and Alarming*. Ph.D. thesis, TU Dortmund University, Germany, Mar. 2012.

[Li11] C. Li, Z. Wang and C. Yang. *Secure Routing for Wireless Mesh Networks. Int. Journal of Network Security*, volume 13(2), 2011.

[LIB] *Netlink Protocol Library Suite (libnl)*. [Online]. Available: `http://www.infradead.org/$\sim$tgr/libnl/`. Accessed: 2014-09-01.

[Lic10] B. Lichtensteiger, B. Bjelajac, C. Müller and C. Wietfeld. *RF Mesh Systems for Smart Metering: System Architecture and Performance*. In *EEE International Conference on Smart Grid Communications (SmartGridComm)*. Oct. 2010.

[LIN] *Linux Wireless*. [Online]. Available: `http://linuxwireless.org`. Accessed: 2014-09-01.

[Lin12] H. Lin, J. Ma, J. Hu and K. Yang. *PA-SHWMP: A Privacy-Aware Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s Wireless Mesh Networks. Springer EURASIP Journal on Wireless Communications and Networking*, volume 2012(69), 2012.

[Liu08] J. Liu, X. Ye, J. Zhang and J. Li. *Security Verification of 802.11i 4-Way Handshake Protocol*. In *IEEE International Conference on Communications (ICC)*. May 2008.

[Mac98] J. Macker and M. Corson. *Mobile Ad Hoc Networking and the IETF. ACM Mobile Computing and Communications Review*, volume 2(1), Jan. 1998.

[Mah13] M. Mahmoud and X. Shen. *A Secure Payment Scheme with Low Communication and Processing Overhead for Multihop Wireless Networks. IEEE Transactions on Parallel and Distributed Systems*, volume 24(2), 2013.

[Mar02] M. Marina and S. Das. *Routing Performance in the Presence of Unidirectional Links in Multihop Wireless Networks*. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. Jun. 2002.

[MAT11] *MATLAB and Statistics Toolbox Release 2011b, The MathWorks, Inc.*, Aug. 2011.

[Mat14] R. Matam and S. Tripathy. *Provably Secure Routing Protocol for Wireless Mesh Networks. Int. Journal of Network Security*, volume 16(3), 2014.

[Men96] A. Menezes, P. van Oorschot and S. Vanstone. *Handbook of Applied Cryptography*, chapter 13, p. 556. CRC Press, 1996.

[Mer78] R. C. Merkle. *Secure Communications over Insecure Channels. Communications of the ACM*, volume 21(4), Apr. 1978.

[Mer79] R. Merkle. *Secrecy, Authentication, and Public Key Systems*. Ph.D. thesis, Stanford, USA, Jun. 1979.

[MES] *FIPS 140-2 Validation Certificate*. Mesh Dynamics. [Online]. Available: `http://www.meshdynamics.com/documents/MD_FIPS_CERTIFICATION.pdf`. Accessed: 2014-09-01.

[Mil07] B. Milic and M. Malek. *Analyzing Large Scale Real-World Wireless Multihop Network. IEEE Communications Letters*, volume 11(7), 2007.

[MIN] *Minstrel Rate Control Algorithm*. [Online]. Available: `http://wireless.kernel.org/en/developers/Documentation/mac80211/RateControl/minstrel`. Accessed: 2014-09-01.

[Mor12] S. Morgenthaler, T. Braun, Z. Zhao, T. Staub and M. Anwander. *UAVNet: A Mobile Wireless Mesh Network Using Unmanned Aerial Vehicles*. In *IEEE Global Communications Conference (GLOBECOM). Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles (Wi-UAV)*. Dec. 2012.

[Mur10] D. Murray, M. Dixon and T. Koziniec. *An Experimental Comparison of Routing Protocols in Multi-Hop Ad Hoc Networks*. In *IEEE Australasian Telecommunication Networks and Applications Conference (ATNAC)*. Oct. 2010.

[Nav08] A. Naveed, S. Kanhere and S. Jha. *Attacks and Security Mechanisms*. In Y. Zhang, J. Zheng and H. Hu, editors, *Security in Wireless Mesh Networks*. Auerbach Publications, 2008.

[Neh14] M. Nehme and M. Sbeiti. *The IEEE 802.11s/i Security Frameworks in INETMANET*. [Online]. Available: `https://github.com/aarizaq/inetmanet-2.0/tree/inetmanet-2.2/src/securityModule`, May 2014. Accessed: 2014-09-01.

[Neu08] A. Neumann, C. Aichele, M. Lindner and S. Wunderlich. *Better Approach to Mobile Ad-hoc Networking (B.A.T.M.A.N.))*. Internet-Draft: http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00. Stream: IETF Network Working Group, Apr. 2008.

[Ng07] P. C. Ng and S. C. Liew. *Throughput Analysis of IEEE 802.11 Multi-Hop Ad Hoc Networks. IEEE/ACM Transactions on Networking*, 2007.

[Ni99] S. Ni, Y. Tseng, Y. Chen and J. Sheu. *The Broadcast Storm Problem in a Mobile Ad Hoc Network*. In *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*. Aug. 1999.

[OLS] *The OLSR.org Story*. Freifunk Community. [Online]. Available: `http://www.open-mesh.org/projects/open-mesh/wiki/The-olsr-story`. Accessed: 2014-09-01.

[OMN] *References of Using OMNeT++ Models in Industrial Research*. [Online]. Available: `http://www.omnest.com/references.php`. Accessed: 2014-09-01.

[OPE] *Open 802.11s: Open-Source Implementation of IEEE 802.11s* . [Online]. Available: `http://open80211s.org/open80211s/`. Accessed: 2014-09-01.

[Paa09] C. Paar and J. Pelzl. *Understanding Cryptography: a Textbook for Students and Practitioners*. Springer, 2009.

[PAS] *The Integration of the Routing Protocol PASER in the INET-MANET Framework*. [Online]. Available: `https://github.com/aarizaq/inetmanet-2.0/tree/inetmanet-2.2/src/networklayer/manetrouting/PASER`. Accessed: 2014-09-01.

[Pat13] S. T. Patibandla, T. Bakker and R. H. Klenke. *Initial Evaluation of an IEEE802.11s-based Mobile Ad-Hoc Network for Collaborative Unmanned Aerial Vehicles*. In *International Conference on Connected Vehicles and Expo (ICCVE)*. Dec. 2013.

[Per99] R. Perlman. *Interconnections (2nd ed.): Bridges, Routers, Switches, and Internetworking Protocols*. Addison-Wesley, 1999.

[Per13] C. Perkins, S. Ratliff and J. Dowdell. *Dynamic MANET On-demand (AODVv2) Routing*. Internet-Draft: https://tools.ietf.org/html/draft-ietf-manet-dymo-26. Stream: IETF Mobile Ad hoc Networks Working Group, Feb. 2013.

[Phi90] K. Phil. *MACA - A New Channel Access Method for Packet Radio*. In *ARRL/CRRL Amateur Radio Computer Networking Conference*. Sep. 1990.

[Pie14] R. D. Pietro, S. Guarino, N. Verde and J. Domingo-Ferrer. *Security in Wireless Ad-Hoc Networks - A Survey. Elsevier Computer Communications*, volume 51(0), 2014.

[Poj11] J. Pojda, A. Wolff, M. Sbeiti and C. Wietfeld. *Performance Analysis of Mesh Routing Protocols for UAV Swarming Applications*. In *International Symposium on Wireless Communication Systems (ISWCS)*. Nov. 2011.

[Pro07] N. Provos and T. Holz. *Virtual Honeypots: From Botnet Tracking to Intrusion Detection*. Addison-Wesley, 2007.

[PRS] *The Galileo Public Regulated Service (PRS)*. [Online]. Available: `http://www.gsa.europa.eu/security/prs`. Accessed: 2014-09-01.

[RAJ] *Rajant BreadCrumb LX5*. Rajant. [Online]. Available: `http://www.rajant.com/wp-content/uploads/2015/01/Rajant_SpecSheet_LX5.pdf`. Accessed: 2014-09-01.

[Ren10] K. Ren, S. Yu, W. Lou and Y. Zhang. *PEACE: a Novel Privacy-Enhanced Yet Accountable Security Framework for Metropolitan Wireless Mesh Networks. IEEE Transactions on Parallel and Distributed Systems*, volume 21(2), 2010.

[Riv78] R. L. Rivest, A. Shamir and L. Adleman. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM*, volume 21(2), 1978.

[ROB] *RoBoard RB-110.* Online. [Available]: `http://www.roboard.com/RB-110.htm`. Accessed: 2014-09-01.

[Roh10] S. Rohde, N. Goddemeier, C. Wietfeld, F. Steinicke, K. Hinrichs, T. Ostermann, J. Holsten and D. Moormann. *AVIGLE: A System of Systems Concept for an Avionic DigitalService Platform Based on Micro Unmanned Aerial Vehicles.* In *IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Oct. 2010.

[Roh13] S. Rohde, M. Putzke and C. Wietfeld. *Ad Hoc Self-Healing of OFDMA Networks Using UAV-Based Relays. Elsevier Ad Hoc Networks*, volume 11(7), 2013.

[San01] C. Santivanez, B. McDonald, I. Stavrakakis and R. Ramanathan. *Making Link-state Routing Scale for Ad Hoc Networks.* In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. Jun. 2001.

[San02] C. Santivanez, B. McDonald, I. Stavrakakis and R. Ramanathan. *On the Scalability of Ad Hoc Routing Protocols.* In *IEEE Conference on Computer Communications (INFOCOM)*. Mar. 2002.

[San05] K. Sanzgiri, D. LaFlamme, B. Dahill, B. Levine, C. Shields and E. Belding-Royer. *Authenticated Routing for Ad hoc Networks. IEEE Journal on Selected Areas in Communications*, volume 23(3), 2005.

[SB14] Y. Saavedra Benitez, J. Ben-Othman and J. Claude. *Performance Evaluation of Security Mechanisms in RAOLSR protocol for Wireless Mesh Networks.* In *IEEE International Conference on Communications (ICC)*. Jun. 2014.

[Sbe11a] M. Sbeiti, T. Tran, S. Subik, A. Wolff and C. Wietfeld. *MuSE: Novel Efficient Multi-Tier Communication Security Model for Emergency and Rescue Operations.* In *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Workshop on Mobile Ad-Hoc Networks for Public Safety Systems (WMAPS)*. Oct. 2011.

[Sbe11b] M. Sbeiti, A. Wolff and C. Wietfeld. *PASER: Position Aware Secure and Efficient Route Discovery for Wireless Mesh Networks.* In *IARIA International Conference on Emerging Security Information, Systems and Technologies (SECURWARE)*. Aug. 2011.

[Sbe12a] M. Sbeiti, J. Hinker and C. Wietfeld. *VLX: A Novel Virtual Localization Extension for Geographical Leash-Based Secure Routing in Indoor Wireless Mesh Scenarios*. In *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. Oct. 2012.

[Sbe12b] M. Sbeiti, J. Pojda and C. Wietfeld. *Performance Evaluation of PASER - an Efficient Secure Route Discovery Approach for Wireless Mesh Networks*. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Sep. 2012.

[Sbe12c] M. Sbeiti and C. Wietfeld. *PASER: Position Aware Secure and Efficient Mesh Routing Protocol*. Internet-Draft: https://tools.ietf.org/html/draft-sbeiti-karp-paser-00. Stream: IETF Keying and Authentication for Routing Protocols, Nov. 2012.

[Sbe13a] M. Sbeiti, C. Vogel, A. Wolff and C. Wietfeld. *ROUTE-O-MATIC: A Comprehensive Framework for Reactive Mesh Routing Protocols*. In *International Conference on Computing, Networking and Communications (ICNC)*. Jan. 2013.

[Sbe13b] M. Sbeiti and C. Wietfeld. *On the Implementation Code of the Secure Mesh Routing Protocol PASER in OMNeT++: The Big Picture: Poster Abstract*. In *ICST International Conference on Simulation Tools and Techniques (SIMUTools)*. Mar. 2013.

[Sbe14a] M. Sbeiti and D. Behnke. *Process Oriented, Secure, and Reliable Emergency Group Communication*. Vodafone Innovation Days. [Online]. Available: `http://www.kn.e-technik.tu-dortmund.de/images/Forschung/CNI_Vodafone.pdf`, Nov. 2014. Accessed: 2014-09-01.

[Sbe14b] M. Sbeiti and C. Wietfeld. *One Stone Two Birds: On the Security and Routing in Wireless Mesh Networks*. In *IEEE Wireless Communications and Networking Conference (WCNC)*. Apr. 2014.

[Sbe14c] M. Sbeiti and C. Wietfeld. *The Agony of Choice: Behaviour Analysis of Routing Protocols in Chain Mesh Networks*. In *ICST International Conference on Ad Hoc Networks (ADHOCNETS)*. Oct. 2014.

[Sbe15] M. Sbeiti, N. Goddemeier, D. Behnke and C. Wietfeld. *PASER: Position-Aware, Secure, and Efficient Routing Approach for Airborne Mesh Networks. IEEE Transactions on Wireless Communications*, 2015. Submitted.

[Sch95] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, 1995.

[Sch14] P. Schwabe. *Graphics Processing Units*. In K. Markantonakis and K. Mayes, editors, *Secure Smart Embedded Devices: Platforms and Applications*. Springer, 2014.

[SEC] *Secure Dynamic Cloud for Information, Communication and Resource Interoperability based on Pan-European Disaster Inventory (SecInCoRe)*. European Research Project (FP7). [Online]. Available: `www.secincore.eu`. Accessed: 2014-09-01.

[Sen13] J. Sen. *Security and Privacy Issues in Wireless Mesh Networks: A Survey. CoRR*, volume abs/1302.0939, 2013.

[Sgo13] A. Sgora, D. Vergados and P. Chatzimisios. *A Survey on Security and Privacy Issues in Wireless Mesh Networks. Wiley Online Library Security and Communication Networks*, 2013.

[Sim82] G. Simmons. *Secure Communications and Asymmetric Cryptosystems*. Westview Press for the American Association for the Advancement of Science, 1982.

[SSL] *OpenSSL*. [Online]. Available: `https://www.openssl.org/`. Accessed: 2014-12-22.

[Šub10] S. Šubik, S. Rohde, T. Weber and C. Wietfeld. *SPIDER: Enabling Interoperable Information Sharing between Public Institutions for Efficient Disaster Recovery and Response*. In *IEEE International Conference on Technologies for Homeland Security (HST)*. Nov. 2010.

[Sug12] I. Sugino. *Disaster Recovery and the Research and Development Policy in Japan's Telecommunication Networks*. In *Plenary Talk at Optical Fiber Communication Conference and Exhibition/National Fiber Optic Engineers Conference (OFC/OFOEC)*. Mar. 2012.

[Tan15] *SafeCurves: Choosing Safe Curves for Elliptic-Curve Cryptography*. [Online]. Available: `http://safecurves.cr.yp.to`, 2015. Accessed: 2015-03-18.

[Tec08] L. Techy, C. Woolsey and D. Schmale. *Path Planning for Efficient UAV Coordination in Aerobiological Sampling Missions*. In *IEEE Conference on Decision and Control (CDC)*. Dec. 2008.

[Tew09] E. Tews and M. Beck. *Practical Attacks Against WEP and WPA*. In *ACM Conference on Wireless Network Security (WiSec)*. Mar. 2009.

[Tho07] R. E. Thorup. *Implementing and Evaluating the DYMO Routing Protocol. Master's Thesis. University of Aarhus. Denmark.*, Feb. 2007.

[Tra10] S. Traboulsi, M. Sbeiti, F. Bruns, S. Hessel and A. Bilgic. *An Optimized Parallel and Energy-Efficient Implementation of SNOW 3G for LTE Mobile Devices*. In *EEE International Conference on Communication Technology (ICCT)*. Nov. 2010.

[Tra11] S. Traboulsi, M. Sbeiti, D. Szczesny, A. Showk and A. Bilgic. *High-Performance and Energy-Efficient Sliced AES Multi-Block Encryption for LTE Mobile Devices*. In *IEEE International Conference on Communication Software and Networks (ICCSN)*. May 2011.

[TRO] *Tropos 7320 Outdoor Mesh Router*. Tropos Networks. [Online]. Available: `https://www.tropos.com/products/tropos-7320-wireless-mesh-router.php`. Accessed: 2014-09-01.

[Ulu12] S. Uludag, T. Imboden and K. Akkaya. *A Taxonomy and Evaluation for Developing 802.11-based Wireless Mesh Network Testbeds. Wiley International Journal of Communication Systems*, volume 25(8), 2012.

[UN] *Global Assessment Report on Disaster Risk Reduction*. United Nations. [Online]. Available: `http://www.preventionweb.net/english/hyogo/gar/2013`. Accessed: 2014-09-01.

[Var08] A. Varga and R. Hornig. *An Overview of the OMNeT++ Simulation Environment*. In *ICST International Conference on Simulation Tools and Techniques (SIMUTools)*. Mar. 2008.

[Var10] A. Varga. *OMNeT++*. In K. Wehrle, M. Gunes and J. Gross, editors, *Modeling and Tools for Network Simulation*. Springer, 2010.

[Vie04] L. Viennot, P. Jacquet and T. Clausen. *Analyzing Control Traffic Overhead Versus Mobility and Data Traffic Activity in Mobile Ad-Hoc Network Protocols. Springer Wireless Networks*, volume 10(4), 2004.

[Wah06] S. Waharte, R. Boutaba, Y. Iraqi and B. Ishibashi. *Routing Protocols in Wireless Mesh Networks: Challenges and Design Considerations. Springer Multimedia Tools and Applications*, volume 29(3), 2006.

[Wan05] X. Wang, Y. L. Yin and H. Yu. *Finding Collisions in the Full SHA-1*. In *International Cryptology Conference (CRYPTO)*. Aug. 2005.

[Wan08] X. Wang and A. O. Lim. *IEEE 802.11s Wireless Mesh Networks: Framework and Challenges. Elsevier Ad Hoc Networks*, volume 6(6), 2008.

[Wan09] J. Wang, B. Xie and D. Agrawal. *Journey from Mobile Ad Hoc Networks to Wireless Mesh Networks*. In S. Misra, S. Misra and I. Woungang, editors, *Guide to Wireless Mesh Networks*. Springer, 2009.

[Wie14] C. Wietfeld and K. Daniel. *Cognitive Networking for UAV Swarms*. In K. Valavanis and G. Vachtsevanos, editors, *Handbook of Unmanned Aerial Vehicles*. Springer, 2014.

[Wol94] W. Wollman and Y. Barsoum. *Overview of Open Shortest Path First, Version 2 (OSPF V2) Routing in the Tactical Environment*. In *IEEE Military Communications Conference (MILCOM)*. Oct. 1994.

[Wol03] T. Wollinger, J. Guajardo and C. Paar. *Cryptography in Embedded Systems: An Overview*. In *Embedded World*. Feb. 2003.

[Wol12] A. Wolff, M. Sbeiti and C. Wietfeld. *Performance Evaluation of Process-Oriented Wireless Relay Deployment in Emergency Scenarios*. In *IEEE Symposium on Computers and Communications (ISCC)*. Jul. 2012.

[Woo03] S. Woon, E. Wu and A. Sekercioglu. *A Simulation Model of IEEE 802.11b for Performance Analysis of Wireless LAN Protocols*. In *Australian Telecommunications, Networks and Applications Conference (ATNAC)*. Nov. 2003.

[Wu06] X. Wu and N. Li. *Achieving Privacy in Mesh Networks*. In *ACM Workshop on Security of Ad-Hoc and Sensor Networks (SASN)*. Oct. 2006.

[Wu08] T. Wu, Y. Xue and Y. Cui. *Privacy Preservation in Wireless Mesh Networks*. In Y. Zhang, J. Zheng and H. Hu, editors, *Security in Wireless Mesh Networks*. Auerbach Publications, 2008.

[Xia12] D. Xia, J. Hart and Q. Fu. *On the Performance of Rate Control Algorithm Minstrel*. In *IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*. Sep. 2012.

[Xia13] D. Xia, J. Hart and Q. Fu. *Evaluation of the Minstrel Rate Adaptation Algorithm in IEEE 802.11g WLANs*. In *IEEE International Conference on Communications (ICC)*. Jun. 2013.

[Xu03] K. Xu, M. Gerla and S. Bae. *Effectiveness of RTS/CTS Handshake in {IEEE} 802.11 Based Ad Hoc Networks. Elsevier Ad Hoc Networks*, volume 1(1), 2003.

[Xu10] H. Xu, X. Wu, H. Sadjadpour and J. Garcia-Luna-Aceves. *A Unified Analysis of Routing Protocols in MANETs. IEEE Transactions on Communications*, volume 58(3), 2010.

[Zap02] M. Zapata and N. Asokan. *Securing Ad Hoc Routing Protocols.* In *ACM Wireless Security (WiSe)*. Sep. 2002.

[Zha05] L. Zhao, R. Iyer, S. Makineni and L. Bhuyan. *Anatomy and Performance of SSL Processing.* In *IEEE International symposium on Performance Analysis of Systems and Software (ISPASS)*. Mar. 2005.

[Zha06] Y. Zhang and Y. Fang. *ARSA: An Attack-Resilient Security Architecture for Multihop Wireless Mesh Networks. IEEE Journal on Selected Areas in Communications*, volume 24(10), 2006.

[Zha08] Y. Zhang, J. Zheng and H. Hu. *Security in Wireless Mesh Networks.* CRC, 2008.

[Zha12] S. Zhao, A. Aggarwal, R. Frost and X. Bai. *A Survey of Applications of Identity-Based Cryptography in Mobile Ad-Hoc Networks. IEEE Communications Surveys Tutorials*, volume 14(2), 2012.

[Zha13a] H. Zhao, E. Garcia-Palacios, S. Wang, J. Wei and D. Ma. *Evaluating the Impact of Network Density, Hidden Nodes and Capture Effect for Throughput Guarantee in Multi-Hop Wireless Networks. Elsevier Ad Hoc Networks*, volume 11(1), 2013.

[Zha13b] S. Zhao, R. Kent and A. Aggarwal. *A Key Management and Secure Routing Integrated Framework for Mobile Ad-hoc Networks. Elsevier Ad Hoc Netowrks*, volume 11(3), 2013.