Efficient, Reliable and Secure Distributed Protocols for MANETs.

Steffen Reidt

Technical Report

RHUL–MA–2009–26

28 January 2010

Department of Mathematics

Royal Holloway, University of London

Egham, Surrey TW20 0EX, England

http://www.rhul.ac.uk/mathematics/techreports

Efficient, Reliable and Secure Distributed Protocols for MANETs.

Steffen Reidt

Thesis submitted to the University of London

for the degree of Doctor of Philosophy

Information Security Group

Department of Mathematics

Royal Holloway, University of London

2009

Declaration

These doctoral studies were conducted under the supervision of Dr. Stephen D. Wolthusen. The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, whilst enrolled in the Department of Mathematics as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Steffen Reidt

May, 2009

Dedication

To Daria

Acknowledgements

I would like to thank my supervisor, Dr. Stephen D. Wolthusen, who consistently provided timely and insightful feedback on my work.

I gratefully acknowledge the financial support of the Royal Holloway University of London which was made possible due to the International Technology Allicance, funded by the Ministry of Defence and the US Department of Defence.

I am also extremely grateful to Shane Balfe for many discussions and collaborations and for proof-reading my thesis. I must also thank Professor Kenny Paterson for his support and insightful discussions.

Additionally, I would like to thank Mudhakar Srivatsa, Dakshi Agrawal, Tal Rabin, Rosario Gennaro, Hugo Krawczyk and Shai Halevi for fruitful collaborations during and beyond my time at the IBM T.J. Watson Research Center.

Finally, I would like to thank my family, and especially my parents Elsbeth and Hans-Jürgen Reidt, for their support and encouragement throughout my education. Especially, I would like to thank my girlfriend Daria Deitermann for her abiding support during my time at Royal Holloway and for helping to proof-read my thesis.

Abstract

This thesis is divided into two parts. The first part explores the difficulties of bootstrapping and maintaining a security infrastructure for military Mobile Ad Hoc NETworks (MANETs). The assumed absence of dedicated infrastructural elements necessitates, that security services in ad hoc networks may be built from the ground up. We develop a cluster algorithm, incorporating a trust metric in the cluster head selection process to securely determine constituting nodes in a distributed Trust Authority (TA) for MANETs. Following this, we develop non-interactive key distribution protocols for the distribution of symmetric keys in MANETs. We explore the computational requirements of our protocols and simulate the key distribution process.

The second part of this thesis builds upon the security infrastructure of the first part and examines two distributed protocols for MANETs. Firstly, we present a novel algorithm for enhancing the efficiency and robustness of distributed protocols for contacting TA nodes in MANETs. Our algorithm determines a quorum of trust authority nodes required for a distributed protocol run based upon a set of quality metrics, and establishes an efficient routing strategy to contact these nodes. Secondly, we present a probabilistic path authentication scheme based on message authentication codes (MACs). Our scheme minimises both communication and computation overhead in authenticating the path over which a stream of packets travels and facilitates the detection of adversarial nodes on the path.

Contents

List of Figures

LIST OF FIGURES

List of Tables

List of Algorithms

Abbreviations

AES	Advanced Encryption Standard
BDDH	Bilinear Decisional Diffie-Hellman
CA	Certification Authority
CH	Cluster Head
CMM	Coalition Mobility Model
EBNF	Extended Backus-Naur Form
IBE	Identity-Based Encryption
ID-PKC	IDentity-based Public Key Cryptography
IDS	Intrusion Detection System
ISB	Incident Shadow Boundary
KAS	Key Agreement Scheme
MAC	Message Authentication Code
MANET	Mobile Ad hoc NETwork
MOCA	MObile Certificate Authority
PGP	Pretty Good Privacy
PKC	Public Key Cryptography
PKI	Public Key Infrastructure
RPGMM	Reference Point Group Mobility Model
RSB	Reflection Shadow Boundary
RSSI	Received Signal Strength Indicator
TA	Trust Authority
VANET	Vehicular Ad hoc NETwork

Chapter 1

Introduction

Contents

In this chapter we provide an overview of the thesis as a whole. We discuss the motivation for our research and describe the contributions of this thesis.

1.1 Motivation

Network security has been an extensively studied field of research for over 40 years, but still raises new possibilities and challenges. Today's mobile phones, laptops and even cars can be equipped with network hardware that allows any of them to directly communicate with other devices. The resulting networks are mobile, decentralised and appear "ad hoc", hence the moniker *Mobile Ad hoc NETworks* (MANETs).

A Mobile ad hoc Network is an autonomous network comprised of free roaming nodes which communicate wireless by radio transmission. MANETs are already ubiquitous and their range of use will spread in the near future. For example, car-to-car communication will allow up-to-date traffic information exchange, informing a car about a nearby accident at the moment of impact. Additionally, emergency

response and military organisations are promising future avenues for this technology. However, all these new possible deployments enabled by MANETs come at the risk of an insecure wireless communication and thus with the challenge to provide algorithms for secure and reliable communication on resource-constrained devices.

One of the main challenges in MANETs is the design of efficient and light-weight security algorithms, that can be handled by devices with limited computational capabilities. Efficiency, reliability and security are (competing) design goals for algorithms suitable for MANETs. Many protocols neglect at least one of these goals: While cryptographic algorithms are typically provable secure and reliable to the extend that lost messages are simply handled with retrials, they marginally consider communication costs. Many state of the art algorithms have a computational and communicational complexity that exceeds the capabilities of resource-constrained MANETs. To overcome these efficiency barriers, new protocols need to be developed that exploit the specific infrastructure as provided by the MANET. Especially pre-configuration of a MANET or a recurring back-link to an infrastructure network may allow the design of more efficient security protocols by equipping the MANET nodes with additional keys or certificates prior to deployment. In contrast to cryptographic protocols, the emphasis of most network protocols such as routing and clustering is not on security. Efficiency and efficacy are the major design goals for these protocols, while security against a plethora of often unpredictable attacks and reliability under abrupt topology changes are up-to-date research problems.

In complex and dynamic networks, the outcome of network protocols such as routing protocols is unpredictable and requires simulations for verification. While network simulators have been used for the simulation of wireless and mobile networks for about 10 years, essential parts such as the mobility modelling are still quite rudimentary. Modelling of the physical layer, which includes mobility and the transmission of wireless signals, is the most crucial part of network simulators.

All mistakes made in the physical layer can be amplified in upper layers and consequently yield wrong results. The development of tools that allow researchers to base their simulations on more realistic simulation scenarios started to gain more attention in the recent years.

1.2 Organisation of thesis and summary of contributions

The remainder of this thesis starts with two introductory chapters. The first of these chapters, Chapter 2, provides an overview of state of the art research in the area of secure protocols for MANETs. In the second introductory chapter, Chapter 3, we extend the network simulator NS-2and define simulation scenarios that are used in the remainder of this thesis. The remaining chapters are divided into two parts. Finally, Chapter 8 concludes the thesis with a discussion on future work.

In Chapter 2 we give an overview of distributed security protocols in MANETs. This chapter provides introductory material for the remainder of the thesis. We identify the category of protocols that we investigate in this thesis, and discuss cryptographic techniques that are available for the design of distributed protocols. We give an overview of existing work, and highlight the specific challenges of distributed protocols for MANETs. Furthermore, we categorise different types of MANETs and identify the specific properties and constraints of MANETs that we study in this thesis.

In Chapter 3 we provide an overview of the techniques used to model wireless transmission in network simulations. We highlight the weaknesses of Open Source network simulators and develop new models to simulate mobility in MANETs and wireless transmission, as well as mobility in urban environments. Urban environments provide the most challenging simulation scenarios since communication links

are likely to be spontaneously interrupted by obstacles. The simulation environment developed in this chapter is used to investigate network protocols in the remainder of this thesis.

Part I:

In Chapter 4 we study the use of cluster algorithms to establish a distributed *Trust Authority* (TA) in MANETs. We investigate security threats of cluster algorithms, and demonstrate that existing cluster algorithms quickly consume the entire battery power of small mobile devices. We modify an existing cluster algorithm, extensively improving its efficiency and making it configurable to suit desired security and efficiency needs. This chapter concludes with an examination on the useability of cluster algorithms in MANETs.

In Chapter 5 we propose two schemes for hierarchical non-interactive key distribution. We prove the resilience of both schemes against a large number of malicious nodes and investigate their feasibility for MANETs regarding computation and communication costs.

Part II:

In Chapter 6 we explore the reliable execution of distributed security protocols under a dynamic network topology. We propose an algorithm that facilitates the efficient and reliable execution of distributed protocols within a given time-frame. Our algorithm determines a quorum of trust authority nodes required for a distributed protocol run based upon a set of quality metrics, and establishes an efficient routing strategy to contact these nodes.

In Chapter 7 we develop a probabilistic path authentication scheme to detect and diagnose routing misbehaviour in MANETs. Its efficiency and short tag size makes

it suitable for MANETs. The scheme builds on symmetric keys whose distribution is analysed in Chapter 5.

The main contributions of this thesis can be summarised as follows:

- We introduce our Coalition Mobility Model (CMM), software for the generation of group mobility files in urban environments for NS-2and other network simulators.
- We develop a cluster algorithm to dynamically bootstrap a distributed trusted authority in MANETs. The novelty of our cluster algorithm is the incorporation of a trust metric, providing robustness against an active adversary.
- We investigate the distribution of symmetric keys in MANETs based on non-interactive key distribution protocols.
- We present a novel algorithm for enhancing the efficiency and robustness of distributed trust authority protocols for MANETs, reducing the communication overhead of small tactical networks (consisting of 50 to 150 nodes) by approximately 32 % over naive broadcast-based approaches.
- We develop a probabilistic path authentication scheme for MANETs, minimising both communication and computation overhead in authenticating the path over which a stream of packets travels while facilitating the detection of adversarial nodes in the path.

1.3 Publications

This thesis contains material that was previously published with S.D. Wolthusen [116, 117, 118], material that is under submission with S.D. Wolthusen and P. Ebinger [114], material that was published with S.D. Wolthusen and S. Balfe [119],

material that was published with M. Srivatsa [115] as well as material that was published with R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin and S.D. Wolthusen [57]. These publications form a basis of Chapters 3 through 7 as follows:

- Chapter 3: [114] and [118];
- Chapter 4: [116] and [117];
- Chapter 5: [57];
- Chapter 6: [119];
- Chapter 7: [115].

Chapter 2

An overview of MANET security

Contents

In this chapter we identify the challenges of designing secure distributed protocols for MANETs, and give an overview of the respective areas of research. Key aspects are the management of cryptographic keys, the provision of a trusted third party and attacks on security protocols.

To secure MANETs against a plethora of often unpredictable attacks, security protocols are required that take the specific constraints and characteristics of MANETs into account. Efficient and light-weight security protocols are needed that can be handled by devices with limited computational capabilities. On the one hand, the requirements on efficiency and security are high, but on the other hand the devices' capabilities and bandwidth provided by the communication channel is limited. To this end, protocols must be designed to optimally exploit the available infrastructure and possibilities for pre-configuration of the respective MANET.

In Section 2.1, we start this chapter by defining *Tactical* MANETs, the type of MANETs this thesis focuses on. We continue with a discussion on symmetric and asymmetric cryptography in MANETs in Section 2.2, followed by an overview of key management in MANETs in Section 2.3. Cryptographic keys form the basis for secure protocols; we discuss the benefits of symmetric versus asymmetric keys in MANET security protocols. MANETs especially lack a central *Trust Authority* (TA), that can be used for key and certificate management. An overview of approaches to overcome this lack of a central TA is given in Section 2.4. Section 2.5

then introduces the concept of secret sharing, which provides the basis for the development of secure distributed protocols, and gives an overview of distributed protocols for MANETs. In Section 2.6 we discuss attacks on cryptographic primitives and distributed network protocols. We conclude this chapter with Section 2.7.

2.1 Tactical mobile ad hoc networks

A MANET, as described by the Internet Engineering Task Force MANET working group, is a temporary or permanent autonomous network comprised of free roaming nodes. The nodes within these networks are wireless communication devices [39] and are typically described by the following characteristics:

- nodes move autonomously resulting in a dynamic network topology;
- nodes may be powered by limited energy source and may have constrained physical security [39];
- messages between nodes are typically routed in a multi-hop fashion;
- communication links between nodes may be bandwidth-constrained.

MANETs can further be categorised by their specific network size (number of nodes in the network), the respective mobility patterns and the capabilities of the mobile devices. In addition to this, dedicated infrastructural elements within a MANET may be not present, ephemerally available, or need to be built from the ground up.

In this thesis we focus on *Tactical MANETs*, which we specify regarding topology, device characteristics and available infrastructure, in the remainder of this section. We primarily deal with military networks but we note that other Tactical MANETs, as can be found in emergency responce settings, have similar characteristics.

2.1.1 Network topology

We assume a Tactical MANETs to consist of 10 to 150 nodes. These networks, as can be found in military and emergency response networks, distinguish themselves due to their structured mobility patterns. In emergency response networks for example, nodes are likely to follow the same paths again and again (bringing people from one rescue station to another, or constantly checking patients' conditions). In military networks, nodes typically move in groups following formations. The crucial differences between the occurring mobility patterns in Tactical MANETs and networks with randomly moving nodes are that they provide a more predictable topology but often a sparsely connected network.

2.1.2 Device characteristics

We assume that Tactical MANETs primarily consist of mobile-phone-sized devices that can easily be carried by humans. State of the art handhelds such as the iPhone[1] have a processing power of approximately 700 MHz. State of the art military handheld-sized devices are still limited to radio frequency phones[2], but as soon as MANETs are secure and reliable enough, commercial products can be adapted to military applications to allow data exchange including video material between handhelds. Second class of devices used in cars and tanks are laptops[3] that can communicate with handheld devices via radio communication, and potentially have a back-link to an infrastructure network.

We assume that the devices carried by humans are equipped with omni-directional antennas and have a propagation power that is limited to 100 mW, thus limiting

[1] http://www.apple.com/iphone/
[2] http://www.rfcomm.harris.com/7800V/
[3] http://www.rfcomm.harris.com/7800I/

the communication range to 100 m [4]. Gerharz *et al.* [59] have shown how interference effects can be minimised by assigning appropriate individual transmission powers to devices in a MANET. Such methods can help to optimise data throughput in communication-intensive protocols such as routing between several communication partners. However, we assume that all devices use the same transmission power, as we are not focusing on communication-intensive protocols in this thesis.

We assume that nodes communicate in a bandwidth between 2 GHz and 5 GHz. The battery of an iPhone allows 6 hours Wi-Fi Internet use, 7 hours of video playback or 24 hours of audio playback. As the duration of a mission might exceed a few hours or a day, it becomes obvious that security protocols need to keep the use of both computation (as represented by video and audio playback) and communication intensity as low as possible.

2.1.3 Network infrastructure

A primary assumption, that substantially influences the design of cryptographic protocols, is the existence/absence of a back-link to a dedicated infrastructure. We therefore categorise MANETs in the following classes:

- **Self-organised MANETs without pre-configuration** These MANETs have no back-link to an infrastructure network whatsoever and no pre-shared keys or any other pre-established infrastructure. The nodes in the network come together as a group of strangers (for a common purpose), and from thereon establish trust relationships, keys and all necessary security associations and infrastructure. While these MANETs are the most challenging ones, tactical networks or any other networks where security is a major issue, will typically not be deployed without any pre-configuration. This thesis therefore

does not deal with this kind of MANET.

- **Self-organised MANETs with pre-configuration** These networks have no back-link to an infrastructure network whatsoever, but are to some extent pre-configured to accomplish a certain operation. The nodes in the network have operational acquaintances and trust each other from the outset, possibly share keys and benefit from an additional pre-configuration. This kind of network presents one typical class of Tactical MANETs. For example, imagine a military scenario where a platoon is deployed in an area without dedicated infrastructure. The mobile devices used by the soldiers in the platoon will be pre-configured at the base, but from the time of deployment, a back-link might technically not be possible or not permitted for security reasons . Self-organised MANETs with pre-configuration are therefore a type of network that we will focus on in this thesis.

- **Back-link-supported MANETs** These networks have a permanent or recurring back-link to an infrastructure network. Nodes can be pre-configured at the beginning of an operation in the same way as for self-organised MANETs with pre-configuration. Furthermore, a permanent or recurring back-link can be used to refresh keys or adjust the network configuration. Back-link-supported MANETs (especially with a recurring back-link) are also considered in this thesis.

2.2 Symmetric and public key cryptography in MANETs

Symmetric and public (asymmetric) key cryptography provide a huge variety of protocols, e.g., for encryption, signatures and authentication, which are suitable for different applications due to their specific requirements . In Section 2.1 we have defined the specific constraints and characteristics of Tactical MANETs. In this

section we discuss the use of symmetric and public key cryptography in MANETs.

2.2.1 Symmetric key cryptography

In symmetric key algorithms, two or more parties need to share a common key with a size of typically 128 bits or more. When a party wants to send a message that only the owners of this key can read, it encrypts the message either bit by bit using a *stream cipher*, or it encrypts the message in blocks of fixed size (e.g., 128 bits) using a *block cipher*. The primary advantage of symmetric key algorithms is their efficiency. The fact that hardware implementable bitwise XOR and AND operations are used for encryption and decryption, makes symmetric key algorithms suitable for devices with very limited computational capabilities. The major drawback however is the requirement for a shared key. This might either be critical due to the lack of a secure method to exchange such a key, or when the number of keys required exceeds a devices' storage capabilities.

Symmetric key cryptography in Tactical MANETs Once shared keys have been exchanged and stored, symmetric key cryptography is the desired choice to encrypt/decrypt data in networks with limited computational capabilities. As discussed in Section 2.1, we assume small mobile-phone-sized devices for Tactical MANETs which are indeed constrained in their processing power. The major issue in using symmetric key algorithms for Tactical MANETs is the storage of the keys and the exchange of the keys in the absence of a trusted authority.

2.2.2 Public key cryptography

In public key cryptography, also known as asymmetric cryptography, the key used to encrypt a message differs from the key used to decrypt it. In public key cryptography, a user has a pair of cryptographic keys, a public key and a private key. The public key can be distributed freely, while the private key is kept secret. Messages are encrypted with the public key, and can consequently be encrypted by everyone. Only the entity in possession of the corresponding private key can decrypt the message.

Private and public key pairs are mathematically related, but it is computationally impossible to derive the private key from the public key. The mathematical techniques required to fulfil such properties include multiplications and modulo operations of numbers that are too big to be factorised. The time required for these multiplications and modulo operations on a state of the art laptop is in the range of milli-seconds or fractions of milli-seconds. An extensive use of public key cryptography in an algorithm can therefore quickly impose computation times of several seconds. The big advantage of public key cryptography compared to symmetric key cryptography is that shared keys are not required. However, the computationally expensive operations restrict the use of public key cryptography to devices with sufficient computational capabilities.

Besides encryption and decryption, public key cryptography can be used for publicly verifiable digital signatures. If a node signs a message with its private key, each node knowing the public key can verify the authenticity of the signature. This concept of public verifiability is a useful feature of public key cryptography that cannot be realised with symmetric key cryptography.

Public key cryptography in tactical MANETs The fact that public key cryptography imposes a critical computational overhead to mobile-phone-sized devices,

as used in Tactical MANETs, does not mean that it should be ignored. Firstly, there is no known way to realise algorithms such as publicly verifiable signatures with symmetric key cryptography. Secondly, combinations of public key and symmetric cryptography might facilitate more efficient algorithms than pure public key or symmetric key solutions alone. An example is the one-time generation of a shared key with public key cryptography (see Section 2.3), which is then used to run symmetric key algorithms. Thirdly, the capabilities of batteries and processors will continue to increase in future, allowing more complex computations on mobile devices. We therefore consider public key cryptography as a suitable, albeit carefully used, operation in Tactical MANETs, while symmetric key cryptography is the choice for frequently repeated and real-time computations.

2.3 Key management in MANETs

As discussed in Section 2.2, symmetric key algorithms are computationally very efficient and are therefore of high interest for MANETs. However, as previously stated, the major challenge in using symmetric key cryptography in MANETs is the secure exchange and efficient storage of symmetric keys. Any data exchange over a wireless channel is initially unauthentic, making it almost impossible to exchange a key without a back-link or pre-configuration. In this section we discuss these issues and give an overview of current research.

2.3.1 Online key exchange

Imagine two parties A and B in a wireless network that want to securely communicate which each other, and for this purpose establish a shared secret key. We assume that A and B are not strangers to each other, i.e., they have some association with

the other party's identity that goes beyond the radio signal they receive.

A protocol that facilitates this pairwise key exchange over an insecure communication channel is the Diffie-Hellman protocol [121]. Since an insecure channel does not provide authenticity of the communicating parties, this protocol is vulnerable to a man-in-the-middle attack. To avoid a man-in-the-middle attack, the identities of A and B need to be linked to the messages they send. The easiest method to link a message with an identity is to communicate over a secure side-channel, which could be provided by an electrical contact as proposed in Stajano's and Anderson's paper [129].

In the absence of a secure side-channel and without physical contact, A and B can only prove the authenticity of their messages using a third party. This third party can be one mutual "friend" or a TA to which A and B communicate over a secure side channel, or the combination of two "mutual friends". Capkun *et al.* have investigated these different possibilities to establish the required security association between A and B [139, 26]. To provide ubiquitous solutions (also for nodes without "friends") for establishing a security association between two nodes (i.e., to authenticate each others' public keys), the management of a TA in MANETs has been extensively studied in the literature [153, 146, 147, 148, 14, 82]. In Section 2.3.2 we give an overview of TA-based and alternative approaches for public key management in MANETs.

Up until now we have discussed the exchange of pairwise keys between two nodes in a network, which typically establish a shared key when required. Some protocols, however, assume shared keys between many or all pairs of nodes in the network. In this case it is more efficient to use a global scheme that distributes shared keys between any pair of nodes with minimal communication overhead. In Section 2.3.3 we give an overview of a number of such schemes proposed in the literature.

2.3.2 Public key management

In the example of two nodes A and B who want to exchange a symmetric key, we have seen the need for a linkage between a public key and an identity. This linkage (also known as a security association) can be provided by a certificate, which proves that a certain public key belongs to a given identity. In this section we give an overview of techniques to issue certificates in MANETs. Most of these techniques go back to the management of a certification authority (CA), which is a trusted entity assigned to manage all certificate issues.

Central certification authority If a back-link to a trusted infrastructure network exists ("Back-link-supported MANETs" from Section 2.1), this back-link provides the secure communication with a central TA. In this case, the management of the CA is not an issue of the MANET but can be provided offline by a traditional trusted authority.

Partially distributed certification authority One of the first approaches to solve the key management problem in MANETs was proposed by Zhou and Haas [153]. Zhou and Haas designed a distributed CA for MANETs, where the power of performing security critical computations is distributed between a set of nodes by letting the nodes share the system secret. The distributed CA signs a certificate by producing a threshold group signature (see Section 2.5.1 for threshold secret sharing schemes). Each server generates a partial signature using its private key share, and submits the partial signature to a combiner. The combiner can be any server and requires at least $k + 1$ shares to successfully reconstruct the digital signature.

The system proposed by Zhou and Haas requires an offline trusted third party to bootstrap the distributed CA. This approach from Zhou and Haas was later extended

by Yi and Kravets [146, 147, 148]. They abandon the need for a combiner and call their CA a *MObile Certificate Authority* (MOCA). While Zhou and Haas did not specify a protocol for the communication of non-CA nodes with nodes from the CA, the MOCA framework concentrates on non-MOCA to MOCA-node communication protocols. Recent improvements of upper schemes include the work from Xu and Iftode [145] and Wu *et al.* [143].

Partially distributed certification authorities provide a promising instrument for "Self-organised MANETs with pre-configuration" (see Section 2.1). The CA can be assigned as part of the pre-configuration to avoid an expensive bootstrapping of the CA within the network.

Schemes to dynamically set up a threshold secret, as typically required by a distributed CA, were proposed by Pederson [101] and later by Gennaro *et al.* [58]. In "Self-organised MANETs without pre-configuration", or in MANETs with pre-configuration, where single nodes of the CA might run out of battery power, there is a need to dynamically bootstrap a CA. The dynamic bootstrapping of a trusted authority, that can act as a CA in MANETs, is discussed in Section 2.4.

Fully distributed certification authority Kong *et al.* [78], Luo *et al.* [87] and Joshi *et al.* [73] proposed public key management solutions, based on the approach originally presented by Zhou and Haas [153]. They distribute the CA over the whole network, i.e., all nodes can act as CA nodes.

The challenges in designing a fully distributed CA are similar to those for partly distributed authorities. Depending on the capabilities of the nodes and the topology of the network, either a partially distributed or a fully distributed approach can be the better choice. In a fully distributed CA, the chance to contact a required number of CA nodes is higher than in a partly distributed CA. Some of the CA nodes might

be several hops away, imposing a higher communication overhead to obtain service than using a partially distributed CA. However, an online bootstrapping of a fully distributed CA imposes high communication costs and is therefore infeasible for larger networks.

A fully distributed certification authority might therefore be favourable in small networks, and when an online bootstrapping of the CA is not required. This can be the case if i) the network can be pre-configured and a later re-establishment of the CA is not required, or ii) the network can be pre-configured and has a recurrent or permanent back-link to an infrastructure network.

A further assumption for a fully distributed CA are homogeneous network nodes, with regard to individual nodes' computational capabilities and trustworthiness. If certain nodes in the network have a higher risk to get compromised or very limited processing power, they should not be part of the CA. In Tactical MANETs, nodes are not necessarily homogeneous; some might be embedded in tanks or carried by soldiers on foot.

Certificate chaining-based key management Capkun *et al.* [27] proposed the concept of *certificate chaining* to manage certificates without a trusted third party. As part of their scheme, nodes issue their own certificates to other nodes and thus do not rely on a centrally managed CA. Each node keeps a limited certificate repository comprising certificates for nodes in its local neighbourhood. When a node wishes to sign the certificate of another node, it simply combines certificate repositories and attempts to find a chain of valid public key certificates between them.

However, apart from the potential benefits of this approach, there is the danger that an attacker can control the signing process by compromising only a small number of nodes. A chain is only as strong as its weakest link, and even a single

compromised node might weaken many certificate chains. As all nodes may be part of certificate chains, this scheme requires that all nodes in the network are equally trustworthy, similar to a fully distributed certification authority.

Identity-based key management Shamir was the first to introduce the concept of *IDentity-based Public Key Cryptography* (ID-PKC) [127] in 1984. However, it took nearly twenty years until an efficient and provably secure Identity-Based Encryption (IBE) scheme was proposed by Boneh and Franklin [24]. By allowing public keys to be derived from a combination of public system parameters and information that uniquely identifies a subject, such as an email address, ID-PKC obviates the need for certificates.

Identity-based schemes require a master secret that must be protected by a TA. This TA acts as the private key generator. Based on the master secret and on input of an identity, it generates the personal private key for a given identity. To make this approach suitable for MANETs, Khalili *et al.* [75] combined identity-based cryptography with a distributed private key generator, i.e., the master secret is protected by a distributed TA. Identity-based key management therefore changes the role of the TA from that of the CA seen in certification authority-based approaches; the TA is not used anymore to certify public keys, but to provide private keys for identities. The security of identity-based key management therefore relies on the security of the distributed TA. Further identity-based protocols such as encryption and signature schemes are discussed in Section 2.5.

2.3.3 Symmetric key agreement protocols

In Section 2.3.2 we gave an overview of the literature on public key management in MANETs. We showed that existing approaches require a (distributed) CA to

certify public keys. Authentic public keys can be used for the exchange of pairwise keys, using the Diffie-Hellman key agreement protocol [121]. Using this method, the establishment of pairwise symmetric keys between each pair of nodes in a MANET would cause an enormous communication overhead. Non-interactive key agreement protocols provide an alternative to equip each pair of nodes with a shared key in a more efficient way. These protocols require pre-configuration or a recurring backlink to set up the required system parameters and to provide each node with a secret key. These schemes are ideally

- *non-interactive:* any two nodes can compute a unique shared secret key without interaction;
- *identity-based:* to compute the shared secret key, each node only needs its own secret key and the identity of its peer;
- *hierarchical:* the scheme is decentralised through a hierarchy where intermediate nodes in the hierarchy can derive the secret keys for each of its children without any limitations or prior knowledge on the number of such children or their identities;
- *resilient:* the scheme is fully resilient against compromise of *any number of leaf nodes* in the hierarchy, and of a threshold number of nodes in each of the upper levels of the hierarchy.

One elegant scheme that has the above first three properties (but weaker security guarantees) was proposed by Blundo *et al.* [22], following the earlier work of Blom [21]. The work of Blundo *et al.* [22] mainly deals with the non-hierarchical setting, but they also discuss an extension to the hierarchical case. In this scheme each node has a secret polynomial (in place of a secret key). A shared key between two leaf nodes is computed by evaluating the polynomial held by one node at a point that corresponds to the identity of the other. An alternative approach to

build a hierarchical scheme was proposed by Ramkumar *et al.* [109], who extended the scheme from Eschenauer and Gligor [50] for the hierarchical case.

Both hierarchical schemes [22, 109] guarantee security only as long as not too many of the leaf nodes are compromised. Once the number of compromised nodes grows above some threshold, an attacker can learn keys of uncompromised nodes, and may even learn the master secret key of the whole system.

A different approach, that is closer to the idea of an identity-based CA, is the identity-based key agreement scheme of Sakai *et al.* [123]. It provides resilience against the compromise of any number of leaf nodes, but it requires a central authority to hand out keys to each and every participant in the network, including any participants joining the network at a later point.

2.3.4 Summary

We have reviewed different techniques for public key management in Section 2.3.2. certificate chaining-based key management appears to be the only technique that does not require a distributed TA. However, this approach requires further investigation to explore under what constraints it provides sufficient security in MANETs, e.g., it is unclear how many chains a certain number of malicious nodes can control. The remaining approaches for public key management require a partially or fully distributed CA. Techniques for the organisation of a distributed TA which can act as a CA are discussed in Section 2.4.

Symmetric keys can either be exchanged by the Diffie-Hellman protocol (if authenticated public keys are in place), or by a non-interactive protocol as reviewed in Section 2.3.3. Such non-interactive key distribution can either be performed during pre-configuration of the network or also requires a TA.

As our discussion in this section has shown, most approaches to key management that are purely performed within the network require a (distributed) TA. The management of a distributed TA induces a significant communication overhead as is discussed in Section 2.4. However, protocols avoiding the use of a TA (such as the Diffie-Hellman protocol for symmetric key exchange) also impose a large communication overhead. For a key management solution that is purely performed within a MANET, a significant communication overhead appears to be unavoidable, and the lesser evil of the existing protocols needs to be chosen according to the requirements of the respective MANET. However, things change, as a MANET is not totally left without external support. Non-interactive key agreement protocols as discussed in Section 2.3.3 give one example of how communication overhead can be minimised by exploiting the possibility of network pre-configuration.

2.4 Bootstrapping a distributed trust authority

Trusted authorities are an essential element in *Public Key Infrastructures* (PKIs) to issue certificates and to manage keys. As our discussion in Section 2.3 has shown, TAs remain an important element for key management in MANETs, even though it is hard to implement a TA in a MANET. The natural approach here is to distribute a TA within the MANET, i.e., to replace the offline TA by an online TA. While protocols for distributed key management were discussed in Section 2.3, this section deals with the actual bootstrapping of the set of nodes that act as the distributed TA in the network.

We focus on TAs that are a subset of all nodes in the network. Using a subset of the MANET as TA is in general favourable over using the whole network as a TA, for two reasons: Firstly, managing a TA with a large number of members might exceed the capabilities of the MANET. This problem was discussed in Section 2.3 in the

context of fully distributed certificate authorities. Secondly, nodes in a MANETs are likely to hold different roles and capabilities, and therefore show a different robustness against compromise.

Choosing a subset of nodes as the TA allows a network to elect the most robust and trustworthy nodes in the network. This subset can either be determined during the pre-configuration phase (if applicable), or can dynamically be established by the nodes in the MANET themselves. Pre-assigning the nodes that form the TA makes the network security dependent on these nodes. If they are compromised or run out of battery power, the security infrastructure of the network is destroyed. However, in certain military scenarios, pre-assignment of a distributed TA might be the best choice. For example, one can imagine soldiers on foot that are supported by some tanks that are in a relative central position. The battery lifetime of the tanks is not an issue, and depending the mission's security on the security of the tanks might be deemed reasonable.

While a dynamic election of the TA members runs the risk of choosing already compromised nodes as TA nodes, the benefits of this approach especially in networks with homogeneous nodes, are:

- TA nodes that run out of battery power can be replaced by other nodes.
- Advantegeous situated nodes (with many nodes within direct communication range) can be chosen as TA nodes to reduce the average cost of other nodes to contact the TA.
- More TA members can be assigned when needed, to allow a network partitioning with two independently functioning TAs.

To react spontaneously to dynamic network changes, the subset of nodes that builds the TA can be re-elected with a certain frequency. Algorithms that undertake the

task of establishing such a subset of nodes are *cluster algorithms*. Typically, cluster algorithms are used to partition the network into clusters, where each cluster is assigned to one *cluster head* (CH). We now give an overview of existing cluster algorithms in the literature.

2.4.1 Cluster algorithms

Cluster algorithms have been widely used in MANETs to determine subsets of nodes for saving energy [36, 33], enhancing routing protocols [7], finding efficient flooding [80, 105], and broadcasting [52], or to generally build low-cost backbones [141]. Clusters have also been applied in recent research on distributing TAs in ad hoc networks [14, 82]. These cluster algorithms build one-hop clusters, i.e., the nodes in a cluster are in direct communication range with their CH. The first cluster algorithm for d-hop clustering was proposed by Amis *et al.* [3].

Bechler *et al.* [14] established a security architecture using clustering and (k, n)-threshold cryptography. In each cluster, exactly one distinguished node, the CH, is responsible for establishing and organising the cluster. Clusters are formed as geographically needed: If nodes cannot find existing clusters, they create clusters themselves, with existing clusters being merged and split on demand.

A major drawback in Bechler's work is the significant relevance of gateway nodes which act as connectors between neighbouring clusters. As Bechler's simulation results illustrate, 34.2 % of the overhead traffic is produced by the gateway nodes, whereas the cluster heads only produce 47.5 % of the overhead traffic, although they incur the management of the security shares.

Conventional clustering is heavily influenced by the initial topology of the network, typically resulting in a central node of the cluster becoming the CH. An ap-

proach that does not take any properties of nodes into consideration, but that assigns cluster heads in a probabilistic way, was proposed by Zongpeng and Baochun [155]. Here, every node participates in a communication backbone with a certain probability dependent on the number of its neighbours. Although this approach is designed to create an energy-efficient backbone, it does not consider the energy and depletion levels of the nodes. Furthermore, the probabilistic assignment of nodes leads to undesirable "bunching" of cluster heads, or leaves large areas without any cluster head (both with certain probability).

From a security perspective, both deterministic and probabilistic cluster algorithms may allow malicious nodes to assign themselves as CHs. In a probabilistic cluster algorithm, such as the one proposed by Zongpeng and Baochun [155], it is impossible to say whether a node cheated to become a CH. All of the reviewed deterministic cluster algorithms choose nodes with the most neighbours as CHs; here nodes could roughly monitor their neighbours' number of neighbours to detect cheating. However, this monitoring mechanism would require additional communication and has not been explored so far. We conclude that none of the existing cluster algorithms in the literature meets the security requirements in MANETs, i.e., remains secure in a meaningful adversary model (see Section 2.6). In Chapter 4 we develop a secure cluster algorithm for the establishment of a distributed trust authority, the development of a provably secure adversary model for this kind of protocol is discussed in Chapter 8.

2.5 Secure network protocols

In this section we introduce the concept of secret sharing, the basic technique for the design of distributed cryptographic protocols. As discussed in Section 2.1, distributed protocols are attractive for MANETs due to the lack of a central trusted

authority. Distributing the power to perform security-relevant computations decreases the risk that a small number of malicious nodes can control the computation. Distributed protocols are mostly used in MANETs for group access control and for signatures. After giving an overview of the concept of secret sharing, we discuss state of the art protocols for group access control and signatures in MANETs. We do not give a concise overview of the theory of secret sharing, rather we focus on the distribution of the secret shares in a MANET, i.e., how secret sharing schemes can be deployed in MANETs. Furthermore, we give an overview of secure distributed networking protocols such as routing and clustering.

2.5.1 Secret sharing

A secret sharing scheme allows a so called dealer to distribute a secret among n parties, where at least $k+1 \leq n$ of the parties need to collude to reconstruct the secret; k or less secret shares do not reveal any information about the secret.

One of the first secret sharing schemes is the (k, n)-threshold scheme proposed by Shamir in 1979 [126]. This approach is based on the property, that a polynomial of degree k can be described by $k+1$ data points.

Protocols for distributed key generation without a dealer, based on univariate polynomials were proposed by Pederson [101] and later by Gennaro *et al.* [58]. These protocols require secure channels between the nodes to submit parts of secret shares secretly, and a broadcast channel for the dissemination of public parameters which are used to validate the correctness of the submitted secret shares. The communication overhead of these schemes, that do not rely on a trusted dealer, is high. Since each of the n nodes needs to send a secret message to each of the other nodes, the communication overhead is at least $O(n^2)$ (if all nodes are within direct communication range). A distributed approach to provide all nodes in a MANET with secret

shares is therefore only suitable for small MANETs. It is in general favourable to distribute secret shares during the pre-configuration phase of a MANET, where a trusted dealer can compute and distribute the secret shares to the nodes.

In MANETs where nodes are likely to join and leave groups, a dynamic admission and revocation of nodes (secret shares) is required. Castelluccia *et al.* [32] proposed a protocol for member admission by $k+1$ secret share holders, i.e., an increase of n without the intervention of a dealer. The scheme is based on sub-protocols from Kong *et al.* [78] and Luo *et al.* [86], which come with the drawback that only one malicious node of the contributing nodes can cause the protocol to produce a useless secret share for the requesting node. The use of a symmetric bivariate polynomial for a (k, n)-threshold scheme was proposed by Saxena *et al.* [125] and Daza *et al.* [43]. If a malicious node contributes a wrong secret share in these protocols, further nodes can be contacted and wrong secret shares can be detected.

While node admission can be performed dynamically and efficiently by the collaboration of $k+1$ nodes, there is no known technique to efficiently revoke secret shares. The only method to truly revoke a secret share is to refresh all secret shares, leaving out the revoked one. This requires the same effort as to initially distribute secret shares, and cannot be performed in an efficient way without a trusted dealer. The same holds for merging and splitting of groups, which can only be established by creating a new secret with new secret shares. More dynamic secret sharing schemes are desirable to facilitate the use of more dynamic distributed TAs in MANETs. In this thesis we restrict ourselves to the use of secret sharing schemes, as this is required to implement a distributed trusted authority (see Chapter 4). The development of more flexible secret sharing schemes is discussed as future work in Chapter 8.

2.5.2 Group access control

Group access control is the direct application of a secret sharing scheme in MANETs. Once a secret is distributed, the secret share holders form a group; holding a secret share means to be a group member. However, group access control is more than secret sharing, it can use additional techniques to organise the group, for example to fix the weakness of inefficient revocation.

Recent papers on group access control have been published by Saxena *et al.* [124, 53, 54], Kim *et al.* [76] and Narasimha *et al.* [94]. The most studied topic in this area is node admission, which can be realised in an efficient way. However, the "headache" of node revocation (as Saxena puts it) is only studied in more depth in [125].

Saxena *et al.* propose in their work [125] to keep membership revocation lists and to validate on each operation whether a node is still an "unrevoked" member or not. Consequently, nodes are technically not revoked, but only written on a "black-list". While this approach avoids the cost for a complete key refreshing, it might increase the risk that the number of malicious nodes reaches $t+1$. At the latest when t nodes are on the membership revocation list, the keys need to be renewed anyway.

Once in place, secret shares of a group can not only be used for distributed computations such as distributed signatures, they can also be used for pairwise key establishment and encryption. Each secret share is issued with a public witness value that allows nodes to validate their secret share's correctness when receiving it. Consequently, the secret share can be used as a private key, and the public witness value as the corresponding public key; this allows nodes to send encrypted messages (with the public key) to the owner of the corresponding secret share. Furthermore, based on their secret shares, each pair of nodes in the MANET can

non-interactively compute a symmetric shared key. This key can be used for secure inter-node communication. For details we refer to [54].

2.5.3 Signatures

Digital signatures allow one party or a group of nodes to sign a message, which can then be verified by other nodes as the signature of this one specific node or of the group of nodes. The group of potential verifiers can either be all nodes, a designated group of nodes or only one specific node.

For MANETs, the crucial properties of signature schemes are the signature length and the required computational effort to sign and to verify a signature. Traditional signature schemes required a signature length of 1024 or 2048 bit to be unbreakable on today's computers. In 2001, Boneh *et al.* [25] proposed the first encryption scheme based on pairings. Pairing-based signature schemes allow secure signatures of 160 bit. Several signature schemes based on pairings have been proposed to meet all varieties of requirements for a signature scheme. A good overview on pairing-based cryptography can be found in "The pairing-based crypto lounge" [12].

Distributed signature schemes suitable for MANETs were introduced by Crescenzo *et al.* [55, 41, 40]. These schemes build on secret sharing as introduced in Section 2.5.1, and adopt the security properties and the (in)flexibility of secret sharing schemes. Distributed signature schemes are consequently secure against k malicious nodes, and the parameter k is chosen fixed for the whole group. As mentioned before, more flexible distributed protocols are desirable for MANETs.

2.5.4 Network layer protocols

So far we have discussed (cryptographic) distributed protocols which are located in the transport layer of the network stack. While in cryptographic protocols security is the major concern, the first design goal of protocols in the network layer (e.g., routing and clustering) is reliability and efficiency. In Section 2.4.1 we already reviewed the literature on cluster algorithms for MANETs and showed that security is neglected in existing cluster algorithms for MANETs. In this section we give an overview on secure routing algorithms for MANETs.

Routing protocols Routing protocols have been extensively studied in the last 10 years, and numerous proactive [102, 35, 37] and reactive [103, 71, 72, 136, 99] protocols were proposed. Furthermore, methods to statistically determine stable paths in routing protocols were proposed [60]. Many of these protocols are reasonably reliable and efficient in specific environments, and even hybrid protocols [107, 133] have been proposed to combine the strengths of different approaches.

While routing protocols have been traditionally optimised for reliability and efficiency, the security of routing protocols has attracted stronger interest in recent years. A routing protocol is secure if an attacker cannot control the process of route establishment. SAODV [150][151] is one of the few routing protocols that provides an example of routing protocol security. It uses hash chains to avoid manipulation of hop counts in route discovery messages, and digital signatures are used for the immutable parts of these messages to provide end-to-end confirmation that the request reached the owner of the address. SLSP [98] is an example of a security mechanism for a proactive routing protocol. It uses signatures on link state update messages to avoid manipulation of the topology information. The SAODV solution is focused on verifying the validity of the path, whereas the SLSP approach is based

on determining the correctness of the network topology. In both cases, the existence of a PKI is assumed.

2.6 Attacks

MANETs provide many points of attack due to the communication over a wireless channel and altering network topology. In this section we first give an overview of attacks on cryptographic protocols. We continue with attacks on keys, including attacks on key distribution and on identities in MANETs. Since distributed trust authorities appear to be a powerful and widely used tool for key and certificate management, we conclude our overview with attacks on distributed trust authorities.

2.6.1 Attacks on key distribution

Keys are the foundation of cryptographic protocols, and an adversary can invalidate the security of the network by holding enough keys. This is especially true for threshold secret sharing schemes as introduced in Section 2.5.1, where the adversary requires $k + 1$ secret keys to take control over security-critical computations. Key distribution protocols must therefore ensure, that keys are only distributed to, or exchanged with, authenticated parties. In Section 2.3 we introduced the Diffie-Hellman protocol [121] for pairwise key exchange over an insecure channel. Two parties who want to exchange a pairwise key do not authenticate each others' identities in this protocol, making the protocol vulnerable to a man-in-the-middle attack. If authentication is required to obtain a key, the adversary's only chance of getting a key is to compromise a node. Once the adversary holds one or several nodes due to node compromise, it will try to gain the most benefit out of its keys. Known attacks that aggregate the power of single compromised keys are the:

- Sybil attack [47];
- node replication attack [100];
- key-swapping collusion attack [92].

In a *Sybil attack*, one attacking node holds multiple identities to gain a disproportionately large influence in the network. An adversary that controls one physical device tries to act as different identities using this one device. In a *node replication attack*, one compromised node is physically copied several times. One stolen identity with respective key material can therefore be used in different physical locations in the network. In a *key-swapping collusion attack*, compromised nodes collaborate to cascade the adversary's impact. Malicious nodes can use keys from other malicious nodes to communicate with good nodes, allowing them to communicate with nodes they share no key with. Furthermore, the malicious nodes can avoid detection by using keys that do not belong to their physical location.

If an adversary has got one or several keys due to node compromise, the good nodes might detect a suspicious behaviour by using an *Intrustion Detection System* (IDS). If a malicious node is detected, there are two ways to deal with it: Firstly, the good nodes could do nothing for the moment, because it might technically not be possible to deactivate single keys; during a later global key refreshing, the malicious nodes could then be excluded. Secondly, a revocation mechanism could be used to dynamically revoke nodes, i.e., deactivate their keys. In this case, an elaborated decision process needs to be implemented in the network that allows good nodes to revoke bad nodes, but if possible not vice versa. A comprehensive survey on key deactivation strategies in MANETs can be found in [10].

2.6.2 Attacks on cryptographic protocols

Today's cryptographic protocols are usually proven to be secure in an adversary model that gives the attacking node(s) a set of capabilities to break the protocol. These models include the *Standard Model* in which an attacker is only limited by time and computational power (expressed by the complexity of a protocol), and the *Random Oracle Model* [15] in which an oracle responds to every query of an attacker with a random and uniformly distributed answer from the possible outputs. Breaking the protocol is then proven to be at least as difficult as solving a well known mathematical problem. Therefore, such protocols are secure within the adversary model as long as the assumptions about used cryptographic primitives hold.

An important primitive in many cryptographic protocols is a hash function, that take a string of arbitrary length as input and produces a pseudorandom output string of defined length. If an attacker can predict the output of a hash function, this might enable him to attack the whole protocol. An attacker will therefore try to find weaknesses of the protocols beyond the security model.

An attack beyond the scope of the protocol is a Sybil attack (Section 2.6.1), that allows one node to have several identities. A single node having $k + 1$ identities in a secret sharing scheme can then control the protocol on its own. The attacks described in the remainder of this section are such attacks that go beyond the scope of cryptographic protocols: Attacks on keys affect the basis of protocols, and attacks on distributed TAs help the attacker to maximise the influence of his nodes. The prevention of these attacks is therefore as important as the security of cryptographic protocols.

2.6.3 Attacks on dynamicly distributed trust authorities

In Section 2.4 we discussed the benefits of a distributed TA in MANETs. If the distributed TA is established dynamically, i.e., during deployment, the members of the TA are determined by a cluster algorithm. Cluster algorithms however base upon communication, and an attacker has manifold possibilities to manipulate this communication. The attacker can:

- replay messages from other nodes to confuse them about their neighbour relationships;
- send messages under wrong identities to influence the choice of TA members;
- cheat about malicious nodes' properties to make them attractive TA aspirants.

Replaying messages has only minor influence on the cluster establishment. Some nodes might connect to cluster heads that are more hops away than expected, but the impact on the choice of the CHs is marginal. Furthermore, the malicious nodes have a high risk of being detected if an intrusion detection system with triangulation for position estimation is used.

Creating messages under wrong identities has direct impact on the choice of the CHs. This attack needs to be prevented by an authentic message exchange, i.e., cluster messages need to be authenticated. As a consequence, we claim that cluster algorithms for security services require authenticated message exchange.

Nodes can cheat about their own properties, for example the number of their neighbours, to be promoted as TA nodes. Cheating about own properties cannot be prevented and is hard to detect.

2.6.4 Adversary models

In Section 2.6.2 we have mentioned adversary models that are commonly used to prove security of cryptographic protocols. Adversary models reflect certain capabilities of an attacker, and a protocol is secure in an adversary model if it can resist any attack within the defined attacker's capabilities. Adversary models therefore provide a framework for clearly defining the security properties of protocols, which is crucial in the complex environment of MANETs.

Attackers vary in their capabilities, and an adversary model might contain a certain percentage of different classes of attacker. The concrete categorisation of attackers depends on the respective protocol. General classes of attacker are passive attackers, active attackers and Byzantine attackers. In an adversary model, the adversary is an abstract entity that controls a certain number of attacking nodes. We will stick to these terms, i.e., an adversary is the abstract entity and an attacker or attacking node is the physical entity that runs the attack. We give a rough categorisation of adversaries that needs to be refined depending on the respective protocol.

- A *passive adversary* (also called honest-but-curious) will only eavesdrop on the network communication.
- An *active adversary* may use the corrupted nodes to prevent the normal functioning of the network via snooping, dropping, modifying, and/or fabricating network messages. Nodes that are actively involved in such attacks and the corresponding faults are called *malicious* or *Byzantine*.
- A *combined adversary* controls a number of nodes that only eavesdrop as well as another set of nodes that runs active attacks.

A further categorisation of the adversaries might be required due to different node capabilities. An assumption that holds in many MANET environments is that attacking nodes have the same computational and communicational capabilities as the honest nodes. Unless otherwise defined, we assume in this thesis that attackers have the same capabilities as honest nodes; in particular, we think of malicious nodes as compromised nodes which consequently have the same capabilities as honest nodes.

2.7 Summary

In this chapter we have given an overview and discussed the challenges of designing distributed protocols for MANETs. We started by defining the specific type of MANETs we focus on in this thesis, so called Tactical MANETs, which have high security requirements but typically benefit from pre-configuration. We have identified the characteristics of symmetric and asymmetric key cryptography which facilitate the development of efficient protocols suitable for power-constrained devices. The distribution of symmetric and asymmetric key material is a major issue in MANETs; we have given an overview of existing key distribution techniques and have shown the importance of distributed trust authorities. After an overview of distributed trust authorities, we introduced the concept of secret sharing which provides the basis for the development of secure distributed protocols. Finally, we have reviewed the most important attacks in MANETs.

Chapter 3

Simulation environment for Tactical MANETs

Contents

In this chapter we discuss the challenges in developing simulators for MANETs. We introduce two extensions to the physical layer of the network simulator NS-2: a lightweight ray optical radio propagation model and a group mobility model. We furthermore define simulation scenarios that are used in Part I of the thesis to investigate and validate network protocols.

Simulations are an important tool to evaluate the performance and reliability of network protocols in MANETs, where topology changes and their impact on protocols are unpredictable. Ongoing changes in communication protocols (e.g.,

802.11) and new capabilities of network devices require the continuous adaptation of network simulation tools.

The main challenge in developing network simulators is modelling the physical layer. As soon as the physical layer is simulated, protocols from all higher levels can be correctly implemented in exactly the same way as they are implemented on real devices. Results from network simulators have to be handled with care. The simulation of the physical layer can only provide an approximation of reality. Mistakes that are made in simulating the physical layer may cause amplified mistakes in the network layer, and so on throughout the network stack. Factors that need to be simulated in the physical layer are the movement of the nodes and the transmission of radio waves used for wireless communication. Taken together, these factors yield an approximate model for the physical layer. In Section 3.1 we introduce our extensions of the physical layer in NS-2, which allow us to implement the simulation scenarios defined in Section 3.2. We use these simulation scenarios to validate the efficiency and reliability of the network protocols investigated in Part I of this thesis.

3.1 Modelling the physical layer

As stated above, the main factors that need to be modelled in the physical layer are the movement of the network nodes and the radio wave propagation.

The movement of the nodes in mobile networks is simulated by mobility models. Mobility models are typically separated from network simulators, so that each mobility model can be used for several network simulators. In Section 3.1.1 we discuss why most of the mobility models used in today's simulations are unsuitable to give a good approximation of a military network, and we develop a new group mobility model suitable for Tactical MANETs.

In Section 3.1.2 we introduce our radio propagation model, which is particularly suitable to model urban environments. The interested reader can find a comprehensive overview of network simulators and their usage in [79].

3.1.1 Mobility model

Although network simulators have been an essential element of research in MANETs for about ten years, mobility models are still surprisingly limited, with the most commonly used model being a random waypoint model. As a result, many statements about the behaviour of MANETs that are based on such simulations may be questionable. We develop a suitable mobility model for tactical networks incorporating both environmental constraints and tactical doctrine. While sometimes the argument is made that random mobility provides the worst case scenario for protocols, we claim that one of the worst case scenarios is provided by groups in urban environments. In these scenarios groups are likely to be separated, obstacles may abruptly cut communication links, and high node densities may cause an overload of the wireless communication channel. In this section we introduce our *Coalition Mobility Model* (CMM), and define simulation scenarios showing the capabilities of the CMM in Section 3.2.

3.1.1.1 Background

Research in mobility models has resulted in a number of models ranging from probabilistic to completely deterministic. Random mobility models represent (almost) probabilistic models since the movements of the nodes is only bound to a few parameters such as the variance of a Gaussian distribution or some constraints which keep the nodes in a bounded area; see [28] for a survey and simulation-based comparison of random mobility models, [16] for a concise categorisation of mobility models in

general, and [6] for a good recent survey on mobility models in tactical networks.

One of the most utilised probabilistic models is the *Random Waypoint Model* [95, 17], in which nodes trace positions which are determined by a uniform distribution. Since the nodes in this model use the shortest path to reach their destination, node density in the centre of the simulation area tends to be higher than in marginal regions. The *Random Direction Model* [122] attempts to avoid this behaviour by sending the nodes on a "detour" via the border of the simulation area.

All of these random models are configurable by few parameters such as the variance of the Gaussian distribution and provide basic mobility patterns for network simulators. A more deterministic movement strategy is provided by the *Graph Model* [135], which restricts the nodes to move randomly on predefined trails. Extensions of this model are commonly used in mobile *Vehicular Ad hoc NETworks* (VANETs), where the nodes (cars) are stopping at cross-ways to simulate traffic lights [106] or move smoothly through curves to simulate bends in the road [16]. Recently, two easy-to-use VANET mobility generators have been implemented [104], [13], which facilitate the automatic generation of VANET mobility files.

A topography-aware mobility model was proposed by Jardosh *et al.* [69, 70]. In Jardosh's *Obstacle Mobility Model*, buildings are modelled as polygons, and the transmission between two nodes is interrupted or highly attenuated if their line-of-sight is intersected by a polygon. The nodes are either allowed to walk on predefined trails or reach their randomly defined aim by the shortest pathway through the obstacle area. An elaborated mobility model for desaster area scenarios was proposed by Aschenbruck *et al.* [5]. Their model supports heterogeneous area-based movement on optimal paths avoiding obstacles with joining and leaving nodes. Aschenbruck *et al.* show how packet loss and data throughput is influenced by heterogeneous node mobility.

Certainly, the most realistic mobility model is one that directly reflects real movements from mobility traces as proposed by Tuduce and Gross [137] and by Lu *et al.* [84]. Their models use traces that are taken from real movements, transferring them in a mobility file which can be processed by the according network simulator. However, generating these traces is very expensive and restricts the simulations to some available trace files.

All previously described mobility models treat nodes independently and thus do not provide any group movement. A generalisation of these models are group models, in which every node moves relative to the logical centre of a group, while the movement of this logical centre can be provided by any of the models above. Consequently, group mobility models need to handle both the movement of the group centre and inter-group movements. They are therefore harder to implement and less well-studied so far.

The first group mobility model for MANETs, the *Reference Point Group Mobility Model* (RPGMM), was proposed by Hong *et al.* [65] in 1999. In the RPGMM, each group has a logical centre and the nodes are randomly but uniquely distributed, moving around the group centre. Wang and Li [140] extended the RPGMM to their *Reference Velocity Group Mobility Model*, in which the movements of the nodes in the group are dependent on each others' velocities. Blakely and Lowekamp [20] fix the relative positions of the nodes to the group centre in their *Structured Group Mobility Model*.

A first model that allows nodes to change groups was proposed by Biao *et al.* [18], but the relative position of the nodes to their group centre is not discussed. Recently, Orchisuren *et al.* [96] proposed an actor-based group mobility model based on RPGMM. In their model, movements of single nodes in the group are influenced by the velocity of the group centre and a random factor that reflects unpredictable

influences on the movement of single nodes. In 2006, Williams and Huang [142] proposed the first group mobility model that combines group mobility and obstacles. They refer to the RPGMM, but use repulsion forces to avoid collusions with other nodes and obstacles.

In all existing group mobility models, the nodes are either in a fix relative position to the group centre or perform random movements within their group. While these approaches are suitable to model groups in which each node moves autonomously within the boundaries of the group, they cannot reflect structured group movements (as in military and emergency response networks and processions). We therefore propose a mobility model that is based on Hong's RPGMM but replace the random mobility within the group with flexible formations.

Several basic implementations, especially of the random mobility models, can be found in network simulators. In this thesis we use the simulator NS-2[51], which offers the possibility to either create totally deterministic movements by writing every single movement directly in the simulation script, or to generate a random waypoint scenario with the script *setdest*. More modular and reusable software for this purpose is provided by the tools *BonnMotion* [138] and *CanuMobiSim* [29]. Both tools are Java-based mobility generators, which provide several random models as well as the possibility to generate mobility files for several common network simulators including NS-2. Moreover, CanuMobiSim provides a graph model, where the graph can either be read from a separate file or directly from an XML file. Due to this functionality, CanuMobiSim was chosed as the basis for our implementantion.

3.1.1.2 Model and implementation

In Tactical MANETs, envisaged in military and emergency response networks, the participants (nodes) are likely to move in groups, which split up, coalesce, and lose

or add single members. As noted in Section 3.1.1.1, a number of random mobility models for pairwise independent node movements have been developed, while the investigation of group mobility models is limited to models that only provide random or no mobility within the group.

In this section we extend the basic idea of the RPGMM and report on a new *Coalition Mobility Model* (CMM), which is designed to be used in conjunction with our topography aware propagation model [114] (both for use on the mobile nodes and to provide more realistic simulations). We illustrate our mobility model using a hierarchicly organised platoon. We note, however, that the mobility model can be used to model any MANET that is organised in one or several groups.

The doctrine for the tactical movements of military formations as described in [61] is to hierarchically organise nodes into one or more formations. Formations are arrangements of soldiers and organised subgroups. Leaders choose formations based on their analysis of the terrain, the likelihood of enemy contact and the need for speed. The smallest group in an infantry operation is the *fire team*. Fire teams typically consist of four soldiers that follow the orders of the team leader. *Squads* form the next group in the hierarchy and consist of fire teams and a squad leader. Squad formations describe the relationships between the fire teams in the squad. Finally, *platoons* present the highest group in this hierarchy and consist of squads in special formations, the platoon leader and other additional soldiers such as the platoon sergeant or a machine gun crew.

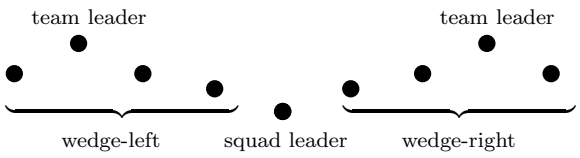


Figure 3.1: Squad in formation “squad line”.

Figure 3.1 shows one possible formation for a squad that is organised as a line. In order to enable the modelling of arbitrary tactical units with changing formations, our implementation of CMM contains a flexible and reusable definition of a group. The entire mobility model, including the groups with their different formations, are defined in an XML file. A group, as considered in CMM is defined in the *Extended Backus-Naur Form* (EBNF) as follows:

```
distance  = "real number"
angle     = "real number"
name      = "string"
node      = distance angle
formation = name {{node} {group distance angle}}
group     = name formation {formation}
```

According to this definition, every node has a fixed desired position in its `formation`, which is described by the `distance` to the `group` centre and the `angle` relative to the direction of the `group` motion. A `formation` itself can also contain complete (sub)`groups` that are also positioned relatively to the `group` centre via `distance` and `angle`. Finally, a `group` contains at least one `formation`. In the case of fire teams, squads and platoons, the `group` "fire team" could contain several `formations` with four `nodes`. The higher-levelled `group` "squad" could then consist of two "fire team" `groups` and an additional `node` as "squad leader", while the highest level `group` "platoon" could consist of `nodes`, "fire team" `groups` and "squad" `groups`.

Finally, for completion of the CMM, the movement of the group centres needs to be defined. We use an extension of the Graph Model, which has already been implemented in the mobility framework CanuMobiSim by Stepanov *et al.* [29]. The nodes in this model are restricted to walk on edges of a connected graph, i.e., there exists a path between each two vertices on the graph. In Stepanov's graph model

[131], every node chooses the next destination vertex uniformly distributed under all vertices, and traces its aiming point on the shortest path. Given that pathways in tactical networks are typically not chosen randomly, and for simulating well-specified scenarios, the routes in the CMM are predefined. Moreover, the CMM supports the consideration of several groups with independent configurations, so that, e.g., several taskforces could walk on different predefined routes. The CMM deliberately does not consider the influence of the topography such as buildings or vegetation. Feasibly complex realisations, such as nodes bouncing on walls or finding the shortest path quoin by quoin are not realistic, while more suitable models tend to be very complex and are subject of ongoing research. Instead we propose the consideration of the topography separately during the simulation calculation. According to a predefined topographical area, the edges of the graph and the group-configurations can be determined manually.

Implementation We have implemented the CMM as an extension of the framework CanuMobiSim [29], which already contains random mobility models and a graph mobility model. An essential feature of CanuMobiSim is the configuration of the respective mobility model in a XML file. We have extended the scope of this XML file to include the description of groups and additional parameters for the CMM. The configuration strategy of a group as defined in EBNF previously allows the re-use of groups of arbitrary depth and thus enables an almost deterministic, but still manageable setup of the CMM. Further extensions of the CMM, such as the changing of nodes between groups or the collection of nodes, can be implemented as required.

3.1.1.3 Summary

We have defined our Coalition Mobility Model CMM that facilitates the generation of formation-based group mobility files for NS-2and other network simulators by the configuration of an XML file. Several hierarchically organised groups can be defined in combination with other mobility patterns as provided by the framework CanuMobiSim. Implementing our model in CanuMobiSim allows an easy extension for further mobility patterns, which are discussed in Section 8.2.1.

3.1.2 Ray-optical propagation model

Even though highly accurate models of radio signal propagation exist, these modelling and simulation environments are of considerable computational complexity and are therefore unsuitable for the incorporation into real-time protocols, particularly on resource-constrained platforms such as MANET nodes. We have therefore proposed a simplified ray-optical signal propagation model in [113] which takes into accound the position of nodes as well as topographical information, but does not incorporate a comprehensive model of physical effects. We have implement the model as a module of NS-2, facilitating an easy integration of our model into NS-2. The core part of our ray-optical propagation model was implemented in the master thesis of Reidt [113]. Improvements regarding efficiency and accuracy were performed in this thesis as well as the validation of the model against real test data. We use the model for the simulations in this thesis; the implemented scenarios are defined in Section 3.2.

3.1.2.1 Background

Hoppe *et al.* [67] introduced a ray optical propagation model that takes into account both reflection and deflection effects on buildings. This model requires the pre-processing of the environment, which by far exceeds the computational capabilities of mobile devices. Pre-processing the data on a powerful server and then storing it on the mobile device is infeasible due to the size of the pre-processed data. The use of this model is therefore restricted to the use of powerful computers or to devices that allow the storage or sending of huge amounts of data is possible. The accuracy of this model was verified in [130] and [111], showing the potential of the approach to model radio propagation by a ray optical model. The model was further extended by Hoppe *et al.* for the use in indoor environments in [66] and [110].

Dhoutaut *et al.* [44] propose the use of the *Shadowing-Pattern Model* to simulate radio wave propagation in VANETs where packet losses occur frequently. This model takes into account most possible types of disturbances while keeping a low computational cost and allowing the easy tuning of any particular disturbance independently of all others. The model is probabilistic and therefore especially useful for VANETs, where disturbance effects are highly correlated with the density of cars, and where typical characteristics of streets allow similar configuration of the model for most VANET scenarios. In Tactical MANETs however, it is difficult to estimate the required configuration parameters of the model. Furthermore, the probabilistic approach cannot take disturbances into account, such as the interruption of signals by buildings in a city.

As noted above, current signal propagation models are typically optimised for high accuracy and they are not intended for use in a resource-constrained environment in which computations must be performed within a near-real-time interval. However, in the following we briefly review several models which are widely used

and which partly form the basis for the ray-optical model as described in Section 3.1.2.2. The models discussed here are typically suitable for describing propagation over arbitrary distances and at frequencies ranging from 1 MHz to 40 GHz unless noted otherwise.

The Free Space model [51] assumes a line-of-sight connection between sender and receiver node without consideration for other influences. Based on these assumptions, the model calculates the power transmitted by the direct line-of-sight connection between sender and receiver. The equation for calculating the power P for a distance d is qualitatively given by $P(r) \sim 1/d^2$. The Two Ray Ground model [51] is a direct extension of the Free Space model which also takes ground reflection of radio waves into consideration [8]. It is based on the assumption of horizontally polarised radio waves, and the power P at distance d is qualitatively given by $P(r) \sim 1/d^4$.

The Shadowing model [51] used in the NS-2network simulator includes line-of-sight components and time-dependent parasitics and scattering. The equation for calculating the receiving power is qualitatively given by $P(d) \sim 1/d^\beta \cdot X$, where $0 < \beta \in \mathbb{R}$ provides a configurable parameter for adjusting the parasitics, and where X is a random variable modelling scattering.

The COST Walfish Ikegami [8] model considers obstacles such as buildings in the vertical plane and effects such as multiple diffraction over rooftops between the transmitter and the receiver node. The transmitter node is assumed to be 4 metre to 50 metre above the ground and the distance between nodes needs to be at least 20 metre. This model is therefore mostly constrained to environments in which the transmitter is located on a rooftop or similarly elevated terrain feature.

3.1.2.2 Model

In the ray-optical propagation model introduced by Reidt [113], a 2D ray-tracing approach is used to develop a simplified but efficient radio propagation model. According to [90] and [56], this approach is defensible under three main conditions:

1. The used frequency band is beyond 1 GHz.
2. Considered surfaces are large in comparison to the wavelength.
3. The surface structures of individual terrain features are approximately constant.

The first condition is satisfied for the ISO 802.11 (a/b/g/h) series of standards (which use bands from 2.4 GHz to 2.5 GHz and 5.15 GHz to 5.85 GHz, respectively). The appropriate wavelength of approximately 10 cm at these frequencies is substantially smaller than the topographic objects such as buildings. Furthermore, the model as specified by Reidt [113] provides only uniform surfaces and does not include additional modifiers such as surface textures. Therefore, it also satisies condition (iii). A further simplification for efficiency is the restriction to vertical surfaces. This simplification allows to store a 2D instead of a 3D map and to use a 2D instead of a 3D raycasting algorithm. In the following section we validate the accuracy of the model of Reidt. A detailed description of the model can be found in [113].

3.1.2.3 Evaluation and analysis

Based on the implementation of our propagation model in NS-2, we now discuss results on the quality of the approximation achieved by the model, as well as empirical data on the performance of the model.

3.1.2.4 Evaluation of calculations

The analysis in [113] has shown the consistency of reflection and deflection factors of the ray-optical propagation with data found in the literature [56]. Beyond this theoretical evaluation, we now report validation results from actual field measurements based on two experiments as reported in [4].

Scenario with obstacles Following the illustration of reflection and deflection factors, which form the main part of the calculations, we next compare the results of our model to measurements from two scenarios. The first scenario in Figure 3.2(a) shows a building and two nodes, representing the sender (dotted circle) and the receiver (crossed circle) [4, Section 3.2.9]. Both sender and transmitter are portable computers equipped with standard 802.11 (a/b) network interfaces. While the sender has a fixed position, the receiver moves away, following the line parellel to the rectangle (building) in this scenario.

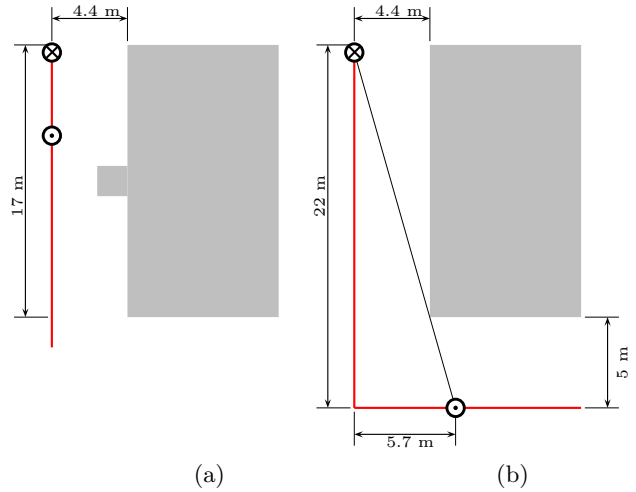


Figure 3.2: Test scenarios.

The power values are measured in the *Received Signal Strength Indicator* (RSSI) [11], so that all power values originally had to be measured in RSSI and transformed to dBm. Unfortunately, there is no standard for transforming RSSI into dBm or

mW. Typically each card manufacturer defines its own relation between RSSI and dBm. This circumstance could be the reason for the almost constant difference of 10 dBm between the measured, and the calculated power values in Figure 3.3 and Figure 3.4. Another reason for this difference could be the ground in the simulation scenarios. While the Two Ray Ground model assumes level ground, the ground surface in the experiments was somewhat uneven and covered with vegetation. As shown in [4], there is a gap of almost 10 dBm between measurements on concrete surfaces as opposed to grass; the reason for this is the different permittivity of concrete and grass. While grass is absorbing much of the transmitted power, concrete and similar substances are reflecting most of it. Moreover, the transmitting power of the sender with a maximum transmitting power of 100 mW was not explicitly defined in [4]. However, we based our calculations on a transmitting power of 100 mW, and compensated for the 10 dBm gap; this gap does, however, indicate the desirability of choosing basic propagation parameters carefully and may indicate a need for incorporating ground permittivity in our constrained model.

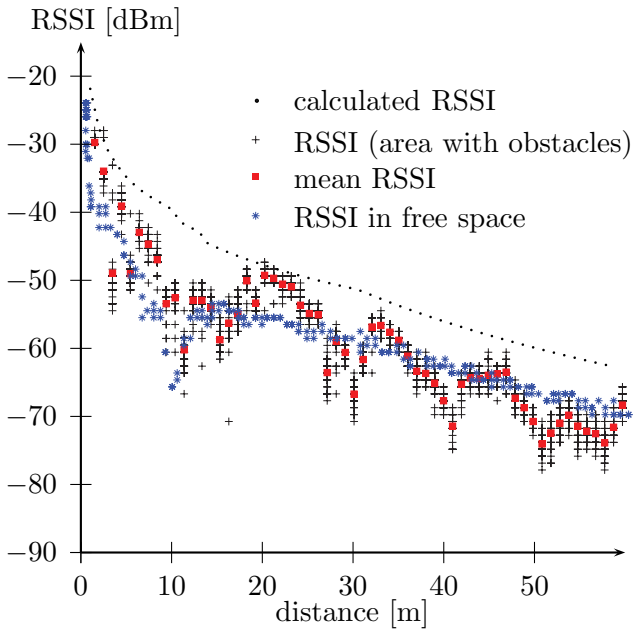


Figure 3.3: Test series 1.

Figure 3.3 illustrates the corresponding power values of the measurements and

the calculation for the first scenario. Each black cross represents one of the measurements, which were done in a distance of 1 m, and the red squares are the mean power values for one distance measured in metres. Additionally, the stars show the results of a measurement done under the same conditions but without any obstacles. The calculations, which were performed using the ray-optical propagation model, are illustrated by the dashed line. Apart from the gap of 10 dBm described above, the curve of the calculated values provides a good fit for the measured values. Owing to simplifications in our model, it is not possible to take interference effects into account. Thus, the curve of the calculated values shows a very smooth behaviour, whereas the measurements show interference patterns, most prominently caused by ground reflection.

Deflection scenario While the propagation in the first scenario was dominated by the direct line-of-sight and the reflection on the ground as well as on the building, the second scenario illustrates the deflection on a house corner (Figure 3.2(b)). While the sender has a fixed position, the receiver is moving behind the building, following the line parallel to the building as before.

As already seen in the first scenario, the curve of the calculated values shows a very smooth behaviour. However, calculated values of our model show a good fit to measured values. After 5.7 m the receiver loses its line-of-sight connection to the sender, resulting in a significant decrease of the receiving power.

The data of the investigated scenarios indicates a high degree of fidelity achieved by our constrained model compared to field measurements. However, we observed that parameters of the underlying Free Space model and Two Ray Ground model need to be chosen carefully. Additional parameters for a future improvement of the model are discussed in Section 8.2.2.

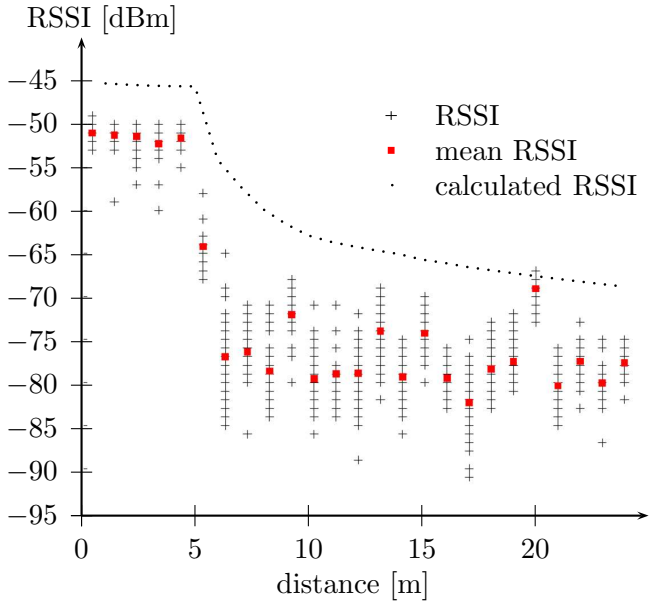


Figure 3.4: Test series 2.

3.1.2.5 Computation Periods

All computations were performed on a Pentium Centrino 1.7 GHz processor with 1 GB of main memory. It should be noted that the resources required for our model including shape file handling do not exceed 5–10 MB depending on the complexity and size of the terrain model.

Figure 3.5 shows the result of a single calculation, which was performed with the help of our iNSpect extension [93]. The scenario shows a 600 m × 600 m square of the centre of London and contains 180 faces and 25 nodes. Such calculations are to be performed on PDAs or other mobile resource constrained devices to improve routing strategies. Table 3.1 lists computation periods based on the scenario described above. Although current PDAs perform at 20–40 % of the performance levels of our test system, improvements in equipment and ongoing optimisation of our algorithms and implementation will significantly reduce the computation times exhibited by our proof of concept model. Results on single routes, however, can already be used effectively for improving existing routing strategies.

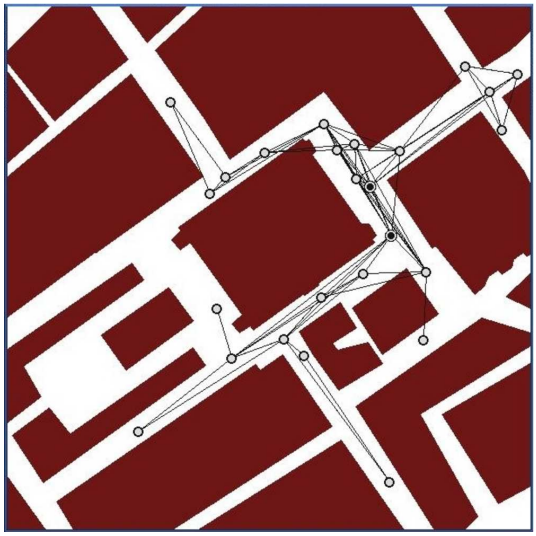


Figure 3.5: Connectivity between nodes.

Table 3.1: Computation periods.

Description	Time [sec]
Power transmitted between two single nodes	0.01
Multihop route with 3 hops	0.025
Multihop route with 5 hops	0.0375
Multihop route with 7 hops	0.05
Connections between all nodes in Figure 3.5	0.6

3.1.2.6 Summary

Based on earlier work [113], we have validated the accuracy and computational efficiency of our ray-optical propagation model which is especially suitable for urban environments. Our evaluation shows the efficiency of our radio propagation model while still obtaining good approximative results. The propagation model is used to simulate urban MANET scenarios, as defined in Section 3.2. Furthermore, it can be implemented on power constrained mobile devices to add valuable information about the connectivity to network protocols, e.g., facilitating elaborated routing protocols and more accurate intrusion detection systems.

3.2 Simulation scenarios

In Sections 3.1.1 and 3.1.2 we have defined an efficient ray optical propagation model and a group mobility model that facilitates the generation of complex group movements including formation changes. In this section we define simulation scenarios that combine these two models to create more realistic simulation scenarios for military networks.

3.2.1 Overview and purpose of simulation scenarios

The purpose of our simulation scenarios is to validate the efficiency and reliability of network protocols in Part I of this thesis. These protocols include the cluster algorithm in Chapter 4 and the distribution of key material in Chapter 5. Based on our mobility model and the ray-optical propagation model, we define three simulation scenarios. Each of the three simulation scenarios contains a platoon of 35 to 37 nodes that accomplishes a certain mission. The communication range is set to 50 m by default, the nodes are represented by soldiers on foot with a speed between 0 m/s and 3.5 m/s. In some of the simulations in later chapters, the communication range might be altered. The mobile devices in the simulation scenarios are supposed to be handheld-sized devices that are equipped with omnidirectional antennas, transmitting at a frequency of 2.4 GHz or 5 GHz according to the ISO 802.11 (a/b/g/h) series of standards.

Simulation Scenario 1 contains a platoon that performs several formation changes, has enemy contact and splits up into two groups to traverse a danger area. The purpose of this simulation scenario is to model a) altering distances between nodes (soldiers) during formation changes, b) splitting and merging of a platoon, and c) failure of single devices during enemy contact. Simulation Scenarios 2 and 3 also

contain a platoon that splits into subgroups in a city (in two different intensities). While the platoon in Scenario 2 splits into several small subgroups that contain at least three soldiers, the platoon in Scenario 3 splits only into three bigger groups (squads) that contain at least 10 soldiers. Scenario 2 represents a worst case scenario for the connectivity of the network, as subgroups are cut-off from the communication to other subgroups at several points. Scenario 3 represents a more realistic mission, in which the platoon only splits in squads but the squads themselves remain as one group. Indeed, Scenario 3 is the result from several discussions about Scenario 2 at military conferences. Since the subgroups (squads) in this scenario traverse parallel streets, nodes from different squads occasionally have eye-contact and the connectivity of the network is better than in Scenario 2. Table 3.2 gives an overview of the benchmarking data of the three simulation scenarios.

Table 3.2: Simulation scenario configurations.

Simulation	Area	#Nodes	Range	Duration	Speed
1	700 m × 900 m	37	45 m	1350 s	0 to 3.5 m/s
2	600 m × 900 m	35	45 m	850 s	0 to 3.5 m/s
3	1200 m × 800 m	37	45 m	1300 s	0 to 3.5 m/s

3.2.2 Application of simulation scenarios

Simulation Scenarios 1 and 2 are used in Chapter 4 to develop a cluster algorithm that provides a reliable cluster according to the number of cluster heads, the frequency of cluster head changes and the proximity of usual network nodes to the next cluster head. Scenario 1 is used to explore the influence of altering distances between nodes, splitting and merging of the platoon and failure of single devices during enemy contact on our cluster algorithm. As our cluster algorithm appeared to easily cope with the topology changes in Scenario 1, we used the “harder” city simulation Scenario 2 to investigate the behaviour of our cluster algorithm under abrupt link breakdowns.

Simulation Scenario 3 is used in Chapter 5 to simulate the distribution of key material. The distribution of the keys without obstacles in Scenario 1 can be performed without problems as the network is connected at all times in this Scenario. Contrary, in our worst case Scenario 2, the distribution of the key material is not possible as the network is never connected. Dissemination of the key material in Chapter 5 is therefore only simulated in Scenario 3.

Further simulations that are based on Matlab[1] are used in Part II of this thesis. However, these are no network simulations that are based on mobility and radio propagation. These simulations will be discussed in the in the respective Chapters.

3.2.3 Detailed description of simulation scenarios

In Simulation 1 (Figure 3.6) a group of 37 nodes traces a route through a hostile area and performs formation changes accordingly. Simulations 2 and 3 contain 35 and respectively 37 nodes, traversing an urban area (Figure 3.7 and Figure 3.8). In all three simulations, the group is organised as a platoon in which the common distance of neighbouring nodes is 10 m. Movement techniques for "travelling", enemy contact and crossing danger areas of platoons were motivated by [61]. As nodes in these simulations are typically represented by infantry on foot, the average speed of the group was set to 2 m/s, while nodes are able to increase their speed up to 3.5 m/s to build up or keep desired formations. We chose transmission power of the nodes to 5 mW, yielding a maximum communication range in free space of approximately 45 m, due to the underlying ray optical propagation model (Section 3.1.2).

Screenshots of the respective simulations are shown in Figure 3.6, Figure 3.7 and Figure 3.8. The nodes are represented by grey and black spots; black nodes represent members of a trusted authority and grey nodes represent non-TA nodes.

[1] http://www.mathworks.com

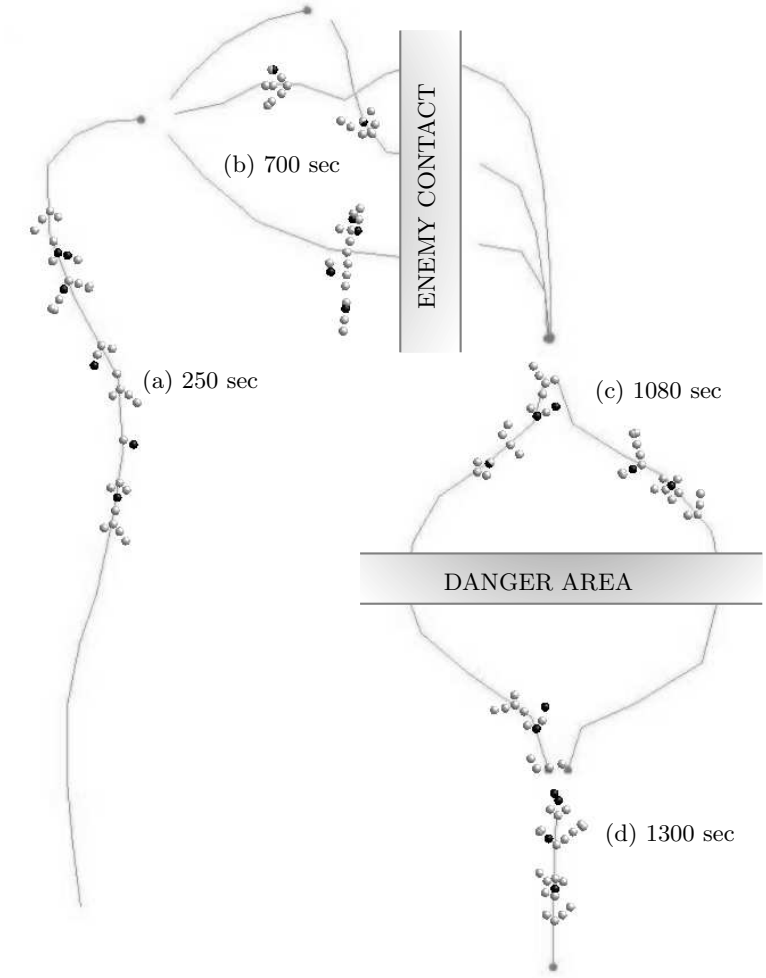


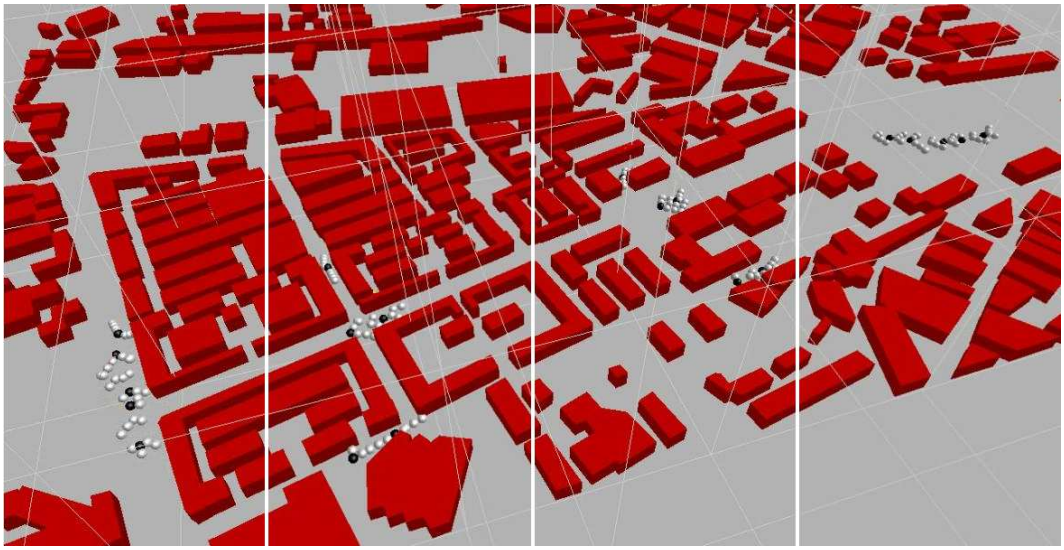
Figure 3.6: Simulation 1: Platoon of soldiers traversing a hostile area.

Simulation scenario 1 Simulation 1 shows a platoon of 37 nodes, first moving in formation "travelling", then having enemy contact and finally passing a danger area. The platoon starts moving at a speed of 2 m/s and stretches while accelerating up to 3.5 m/s (Figure 3.6(a)). Due to an expected enemy contact, the platoon splits up a short time later: two squads follow the lower path while the remaining two squads trace the upper two paths. At second 700, the lower two squads change their formation to a line due to enemy contact (Figure 3.6(b)). During this 120 seconds procedure, the nodes are moving with an average speed of 0.1 m/s and their wireless devices are likely to incur loose contacts or drop out totally. Thereupon, after collating to a platoon again, the group divides to cross a danger area (Figure 3.6(c)) and forms up as a "travelling" platoon again (Figure 3.6(d)).

Figure 3.7: Simulation 2: Platoon of soldiers tracing a city area.

Simulation scenario 2 Simulation 2 shows a platoon of 35 nodes tracing a city area, splitting up in groups of at least three nodes and re-grouping. Figure 3.7(a)) shows the imminent devision of the platoon in three squads after reaching the city area. The nodes have decreased the distances between each other from the typical 10 m to 5 m, yielding a more compact network. 210 seconds later (Figure 3.7(b)), the squads trace independent routes in between the buildings, while several fireteams temporarily leave the squad to occupy further streets. Finally, the squad leaves the urban area and falls back into the original formation (Figure 3.7(b)).

Simulation scenario 3 Simulation 3 shows a platoon of 37 nodes tracing a city area and splitting up in three groups (squads) of 10 to 16 nodes. Figure 3.8(a) shows the platoon in its original formation with the front and the centre squad heading for



(d) 1030 sec (c) 760 sec (b) 315 sec (a) 65 sec

Figure 3.8: Simulation 3: Platoon of soldiers traversing a city area.

the upper street, and the trail squad heading for the southern street. 250 seconds later (Figure 3.8(b)) the three squads have split up. The tail squad remains in the southern street, the centre squad remains in the main street and the front squad with 10 nodes occupies further streets while re-connecting to the centre squad from time to time. Second 760 (Figure 3.8(c)) shows such a situation where the front squad leaves the centre squad to occupy a parallel street in the north. Finally, the squad falls back into the original formation on reaching the end of the streets (Figure 3.8(c)).

3.3 Summary

In this chapter we have proposed two extension to the network simulator NS-2: a ray optical propagation model for the simulation of wireless data transmission and our group mobility model CMM. Our propagation model facilitates modelling inter-node communication in cities where buildings obstruct and reflect radio signals. Our group mobility model allows to simulate group movements including formation

changes and following pre-defined paths such as streets in a city. Based on these two extensions, we have defined three simulation scenarios. Two of the simulations scenarios contain a small military MANET in a city area, one contains a MANET in a hostile area. The simulation scenarios are used in Chapters 4 and 5 to investigate the influence of group partitions and abrupt communication breakdowns on the respective protocols.

Part I

Bootstrapping a security architecture in MANETs

Chapter 4

Bootstrapping a distributed TA in MANETs

Contents

In this chapter we investigate the dynamic bootstrapping of a distributed trust authority (TA) using cluster algorithms. We develop a cluster algorithm that avoids

frequent cluster head (CH) changes, and achieves its security by incorporating a trust metric in the cluster head election process.

4.1 Introduction

The design of a flexible network security architecture presents many challenges in MANETs (see Section 2.4). In traditional network architectures, keys and certificates are distributed and controlled by a trusted central authority. In a MANET, cut off from a hierarchical point of control, such a TA no longer exists. The natural approach to provide an alternative to a central trust authority is to establish a trust authority within the network. On the one hand, assigning a single node as the TA exposes the whole network to get compromised if an adversary takes control over this node. On the other hand, leaving every security critical task to the whole network (by voting or by using distributed protocols), imposes a high communication overhead. A promising tradeoff between security and efficiency can be realised with a *distributed TA*, in which a subset of nodes acts as the TA on behalf of the entire network.

In this chapter we investigate techniques to determine a subset of nodes to serve as a distributed TA. This set of TA "members" can either be static and pre-elected, or deployed online by a cluster algorithm. We show that existing cluster algorithms impose an infeasibly high communication overhead on the network and change the CHs too frequently to maintain a stable TA. We develop a cluster algorithm that fixes these weaknesses and makes TA membership configurable by several metrics for trust, battery capacity, signal strength, bandwidth and reachability. Depending on the IDS's accuracy, the incorporation of a trust metric provides robustness against an active adversary.

4.2 Overview of cluster algorithms and trust metrics

In Section 2.4 we provided an overview on existing approaches to the use of cluster algorithms for the purpose of saving energy, enhancing routing protocols, finding efficient flooding and broadcasting mechanism. Clustering in support of TA services has been studied by Bechler [14] and Jiun [82]. TA "members" are assigned in their proposals based on the number of neighbours. However, this approach lacks security against an active adversary who attempts to control a large ratio of the cluster heads: if the cluster algorithm allows nodes to assign themselves as CHs based on the number of neighbours, the nodes can simply cheat about their number of neighbours to assign themselves as cluster heads.

In this chapter we are interested in developing a *cluster metric* which, if used in cluster algorithms, incorporates the security requirements for a distributed TA. The metric is intended as an open collection of parameters which can be expanded as required; the major parameter however, incorporating the aspect of security, is *trust*. We therefore concentrate on *trust metrics* and constraints imposed by the Tactical MANET environment itself, namely limited battery capacity and radio frequency interface constraints.

Evaluation of the efficacy of the cluster metric is achieved by using the algorithm reported in [3] for max-min d-cluster formation in wireless ad hoc networks. This algorithm results in each node either being a CH itself or being at most d hops away from a CH. The following briefly reviews related work on trust metrics, as this partial metric is the most important one in the context of our system, and has also been the most intensely studied.

Trust Metrics This paragraph reviews a selection of trust metrics which have been proposed in recent years. Since some of these have not been explicitly proposed in the form of metrics, we have adapted them to provide consistent terminology. All models share the use of a digraph-based representation with different vertex and edge valuation interpretations.

One of the first trust metrics was proposed by Zimmermann [154] in 1995. Here, nodes are keys of a public key system and the edges represent certificates. A user assigns a value from the set {*unknown, not trusted, marginally trusted, fully trusted*} to every key he retrieves. The reduction to only four different types of trust allows the model to be easily implemented. However, Kohlas and Maurer [77] showed that due to this simplicity, the model may deliver counter-intuitive results, e.g., that similar chains of trust can lead to different results.

A seminal approach to define a trust metric in the form of a model for public-key certification, trust and recommendations was defined by Maurer [91] in 1996. Maurer established the syntax of *certificates*, *recommendation*, *trust* and *authenticity of public keys*, which form the axioms of his model. Based on these axioms, two intuitive inference rules are defined which permit drawing of transitive conclusions from a set of given axioms. Since the initial model is binary, Maurer inserts the consideration of confidence on a continuous scale between 0 and 1 in a second step. While Maurer's model considers chains of trust of arbitrary length and complexity, it has an exponential complexity in the length of trust chains.

In order to enable a real implementation of Maurer's model and a computation without exponential complexity, Caronni [30] suggested several possible simplifications.

Maurer's model can be considered quite simplistic regarding the choice of axioms. The set of axioms in the original version does not contain a time parameter,

which is necessary for key revocation. Marchesini [88] addressed this issue and extended Maurer's model using axioms for `properties, time` and `domain`, and thus provided numerous additional abilities of the system, including key revocation. Bicakci *et al.* [19] investigated the incorporation of certificate revocation in Maurer's model.

Recent further work on trust metrics includes research by Sun *et al.* [132] who propose two axioms for trust models, namely that (1) concatenating trust values in one path does not increase trust and that (2) multipath propagation of trust does not reduce trust. Sun *et al.* propose two trust models which model trust as a value between -1 and 1. However, both models can return counter-intuitive results, since the concatenation of two negative trust values can result in a positive value in both models. The second axiom of Sun *et al.* is not satisfied in several other trust models. Abdul-Rahman [1] and Xiong [144] calculate the trust value as the average of the values calculated from different paths. According to this, an additional positive but low evidence value will reduce the resulting trust value and thus break Sun's second axiom.

Reiter [120] proposed an efficient trust model that does not require the evaluation of complex trust links. In this model the metric is based on the idea that every chain is only as strong as its weakest link. Thus the trust value representing a node's trust in another node is simply computed by adding the weakest links of all chains of trust between these two nodes. However, this construction can lead to counterintuitive results, e.g., many low trust values can add up to a high trust value.

Recently, Theodorakopoulos [134] proposed an algebraic trust framework. While this approach is mathematically elegant, it remains unclear how to implement the system.

4.3 Assumptions and definitions

In this section we outline the design requirements that our cluster algorithm for TA member selection must satisfy, describe the assumptions that we make about our scheme, outline our adversary model and give definitions used to define our cluster algorithm.

4.3.1 Design requirements

- **Security:** We require that our cluster algorithm is secure against an active adversary as defined in Section 4.3.3.
- **Efficiency:** We require that the cluster algorithm is efficient enough to be run on a MANET node with limited computational and battery capabilities.
- **Reliability:** We require that the cluster algorithm is robust under topology changes and node failures, i.e., provides a set of CHs of almost constant size and rarely changing CHs.
- **Scalability:** We require that the cluster algorithm is scalable with regards to security and efficiency. While a platoon moving in one group through a friendly area should be configured to save energy, security becomes the major concern if enemy contact occurs.

4.3.2 Assumptions

- **Pre-loaded keys:** We assume the presence of a public key infrastructure that facilitates nodes to sign cluster messages.
- **Intrusion detection system:** We assume that an intrusion detection system is active on each node. This intrusion detection system continuously monitors

the behaviour of neighbouring nodes, yielding in trust values (opinions) about other nodes.

4.3.3 Adversary model

We desire a cluster algorithm to be secure against an active (Byzantine) adversary (see Section 2.6.4). Attack possibilities against a bootstrapping mechanism of a distributed TA have been discussed in Section 2.6.3; an attacker can:

- replay messages from other nodes to confuse them about their neighbour relationships;
- send messages using wrong identities to influence the choice of TA members;
- cheat about malicious nodes' properties to make them attractive TA aspirants.

Another critical attack which applies to most network protocols is the Sybil attack in which one node holds several identities (see Section 2.6). We assume a network in which the Sybil attack can be contained, for example by having a cost of entry. An overview of these attacks and prevention techniques in the context of routing algorithms is presented in [2]. To avoid receiving messages from wrong identities, cluster messages need to be authenticated. Each cluster message is therefore signed with the pre-loaded private key of the respective node. The sending of wrong information (cheating) cannot be prevented beforehand and needs to be detected by an intrusion detection system. Replaying cluster messages can be detected (under the assumption of synchronised clocks) by timestamps in the cluster message. If synchronised clocks of the MANET nodes cannot be assumed, an intrusion detection system must be used. The results from the intrusion detection system are the input to our trust metric. Based on gathered evidence both from a node's intrusion detection system

and from neighbour's opinions, our trust metric allows each node to estimate other nodes' reliability and thus their eligibility as a possible cluster head.

4.3.4 Definitions

Ad hoc networks are commonly modeled as a graph $G = (V, E)$, where V is the set of vertices and E the set of edges. In order to investigate the convergence behaviour of our model we propose the following extensions:

Definition 1 (Quality factor) *The* quality factor $r_{ij} \in [0, 1]$ *describes the belief of node i about node $j's$ qualification for being a TA*[1] *member node.*

Definition 2 (Belief Set) *Let V be the set of all nodes, then the relation $R(t) : V \times V \to [0, 1] \subset \mathbb{R}$ contains all* quality factors *of the network at a certain time. R can be identified as a matrix $R = (r_{ij}) \in \mathrm{M}(n \times n; [0, 1] \subset \mathbb{R})$ and is called the* Belief Set.

Definition 3 (TA configuration) *Let s_{ij} denote the TA connection from node i to node j, i.e., $s_{ij} = 1$ if node i chooses the TA member node j to cotact the TA, and $s_{ij} = 0$ otherwise. Furthermore, let $R = (r_{ij}) \in \mathrm{M}(n \times n; [0, 1] \subset \mathbb{R})$ be the* Belief Set*, then a matrix $S = (s_{ij}) \in \mathrm{M}(n \times n; \{0, 1\})$ is called* TA configuration *if:*

$$\sum_{0 \le i \le n} s_{ij} = 1, \quad 0 \le j \le n \tag{4.1}$$

$$s_{ij} = 1 \quad \Rightarrow \quad r_{ij} > 0 \tag{4.2}$$

Thus a subset of nodes is called TA configuration if every node is connected exactly to one TA node.

[1] For consistency, cluster heads are labelled as TA nodes.

4.4 Metric-based cluster algorithm

4.4.1 TA selection mechanism

A cluster algorithm is used to determine the subset of TA nodes in the MANET. The choice of TA connections as utilised by the cluster algorithm does not necessarily establish real TA connection in the later security architecture. In the scope of the cluster algorithm, a node can immediately connect to another node, which may require an interval to establish its state as a TA member. According to this, our cluster-based algorithm for TA member selection provides a sufficient subset of TA nodes, which can then be used for bootstrapping the security architecture on top of these nodes.

Without loss of generality, and as noted in Section 4.2, we use a modification of Amis' algorithm [3] for the initial implementation and evaluation of the metrics used in distributing TA services. The underlying principle of deterministic cluster algorithms is to let each node exchange information with immediate neighbours and to decide whether it is a TA node itself or if it accepts a peer node as a TA node. If a node A accepts another node B as a TA node, node B will be the connector to the TA for node A. In the case of Amis' algorithm, this information exchange procedure is performed d times, yielding a network where every node has a maximum distance of d hops to its TA connector. Amis describes the basic concept of his algorithm as follows:

> Initially, each node sets its *winner*[2] to be equal to its own node id. Then each node locally broadcasts its *winner* value to all of its 1-hop neighbours. After all neighboring nodes have been heard from, for a single round, the node chooses the largest value among its own *winner*

[2] *winner* is a TA node in this context.

value and the values received in the round as its new *winner*. This process continues for d rounds.

Extension 1: Choose TA nodes by quality In our extension of Amir's algorithm, the *winner* value is represented by a quality factor instead of the node identity. Moreover, the base algorithm's approach of choosing its d-hop cluster head based on the decisions of neighboring nodes in round $d-1$ must be augmented since different nodes might hold different views about a node's TA qualification. The base algorithm is therefore extended as follows:

TA Cluster Algorithm: In our cluster algorithm we use the two parameters `hopsToGo` and `forwardInfo`. `hopsToGo` is the remaining number of hops that a message shall be sent. `forwardInfo` is a list containing `hopsToGo` values.

- Each node collects the information broadcasted by neighboring nodes and retains the information until it is refreshed or until it exceeds its predefined lifetime. Cluster information with a `hopsToGo` value greater than 1 are pushed on the stack `forwardInfo`, whereupon the respective `hopsToGo` value is decreased by 1.
- In certain (possibly node-specific) time periods, each node determines all quality factors about its known d-hop neighbours, choosing the node with the highest quality factor as its cluster head. If the node itself holds this value, or if another node has chosen it as cluster head, the node will itself be a TA node. The node then broadcasts its newly determined TA status and its additional information such as the battery level and `forwardInfo` to its neighbours. Every entry of the `forwardInfo` stack contains a parameter `hopsToGo`, which is indicating the number of forwarding hops and initially set to the clustering depth d.

First simulations have shown the tendency of cluster algorithms to change the cluster heads quite frequently. In cluster algorithms for MANETs, nodes broadcast messages in certain frequencies, and after each broadcast clusters are reshaped and cluster heads are likely to change. We observed that even with small topology changes between these broadcast phases, caused by node movements, cluster heads mostly changed. This behaviour could be prevented by configuring the quality factor to assign a higher quality to TA member nodes and thus foster their reelection. However, this would be a misuse of the quality factor that would decrease its potency for more sensitive configuration issues such as energy level and trust values. We therefore augmented the cluster algorithm itself by a mechanism for avoiding abrupt changes of the cluster heads.

Extension 2: Avoid frequent changes of TA members. To avoid frequent changes of TA members, we firstly augmented the possible set of TA states `TA_MEMBER` and `NOT_TA_MEMBER` by `TA_ASPIRANT` and `LEAVING_TA`. The `TA_ASPIRANT` parameter is used to insert a second step into the process of becoming a TA member. Accordingly, a node first changes its state to `TA_ASPIRANT` if it holds the highest quality value within its neighbourhood. After a certain configurable period `CONST_TA_INTEREST` as `TA_ASPIRANT`, the node will become a TA member if it still holds the highest quality value. In our simulations in the following section, the parameter `CONST_TA_INTEREST` was set to three times the cluster message frequency. The respective mechanism was evaluated with the help of the parameter `CONST_NO_TA_INTEREST` for the state `LEAVING_TA` to avoid a abrupt release of the `TA_MEMBER` state. This parameter was set to one time the cluster message frequency, when using greater values it appeared to drastically bar the nodes from leaving the TA.

Summarising our extensions to Amis' [3] algorithm, Algorithm 4.1 shows the protocol specification of our TA cluster algorithm.

Algorithm 4.1: TA cluster algorithm pseudocode.

Input: cluster frequency cf, d, cluster metric
`/* In cluster frequency` cf `node` i `does: */`
Determine quality value r_{ij} for all nodes in d-hop neighbourhood;
if *the node with the highest value* r_{ij} *is a TA member* **then**
  node i chooses this node with the highest r_{ij} value as its TA connection;
else
  i sends 'you are my best quality' message to the node with the highest r_{ij} value (possibly itself);
  i chooses the TA member with highest r_{ij} in its d-hop neighbourhood as its TA connection (possibly none);
end
if *node i got a 'you are my best quality' message in the last period cf (possibly from itself)* **then**
  if hasStatus(`TA_ASPIRANT`) **then**
    `TA_ASPIRANT_TIMER` $+= cf$;
    if `TA_ASPIRANT_TIMER` $\geq$ `CONST_TA_INTEREST` **then**
      setStatus(`TA_MEMBER`);
    end
  else if hasStatus(`LEAVING_TA`) **then**
    setStatus(`TA_MEMBER`);
  else if hasStatus(`NOT_TA_MEMBER`) **then**
    setStatus(`TA_ASPIRANT`);
    `TA_ASPIRANT_TIMER = 0`;
  end
else
  if hasStatus(`TA_ASPIRANT`) **then**
    setStatus(`NOT_TA_MEMBER`);
  else if hasStatus(`LEAVING_TA`) **then**
    `LEAVING_TA_TIMER` $+= cf$;
    if `LEAVING_TA_TIMER` $\geq$ `CONST_NO_TA_INTEREST` **then**
      setStatus(`NOT_TA_MEMBER`);
    end
  else if hasStatus(`TA_MEMBER`) **then**
    setStatus(`LEAVING_TA`));
    `LEAVING_TA_TIMER = 0`;
  end
end

4.4.2 Metrics

The choice of the cluster heads in our algorithm is based on the quality factors. In Definition 1 the quality factor was fixed as a value in the continuous interval from 0 to 1. A quality factor $r_{ij} = 0$ means that a node i has no evidence about a node j, while a value of 1 perfectly qualifies node j as a TA node according to our metrics. In this section we develop several partial metrics, which will be combined to the *cluster metric*.

Each partial metric is mapped onto the continuum $[0, 1]$, assuming no constraints are violated. In the case of a constraint violation of one or more partial metrics, the cluster metric will itself yield 0 and thus disqualify a node as a TA node. The partial metrics are merged into a cluster metric using a linear combination. This itself requires a linear and continuous mapping of the partial metrics and weighting for relative importance. The metrics discussed in this section are not exhaustive; partial metrics can be replaced and additional partial metrics be used as discussed in Section 4.6. The core of our cluster metric is the trust metric, which is derived from Maurer's [91] model for a public key infrastructure. For the remaining metrics for signal strength, energy level, bandwidth and incorporation in the routing process, there exist no elaborated metrics in the literature.

4.4.2.1 Trust metric

The trust metric is the core of the *cluster metric*, since it induces the cluster algorithm to determine a set of essentially trustworthy TA nodes. In our approach, a modification of Maurer's [91] model can be used, containing both a different trust model and valuations, provided that the constraints described in Section 4.4.2 are satisfied. Maurer's model consists of two parts, a deterministic and a probabilis-

tic part. However, the basic model is not suitable for implementation owing to its computational complexity, and must be adapted in its deterministic part as described in the following. We first describe Maurer's model, before we introduce our modifications of Maurer's model.

- **Deterministic part** The deterministic part defines the parameters which are considered by the model, and defines inference rules for these parameters. Maurer labels the parameters as *statements* which include the *Authenticity of public keys*, *Trust*, *Certificates* and *Recommendations*. Based on those statements, Maurer defines two inference rules which consider recommendations of arbitrary depth. For example, if a node A believes in the authenticity of a node X and it also trusts X to administer certificates and X holds a certificate of Y, then A will also believe in the authenticity of node Y (see [91] for details).

  The statements in Maurer's model can be described in more detail as follows:

  - *Authenticity of public keys.* $Aut_{A,X}$ denotes A's belief that a particular public key P_X is authentic.
  - *Trust.* $Trust_{A,X,1}$ denotes A's belief that a particular entity X is trustworthy for issuing certificates. Similarly, her belief that X is trustworthy for issuing recommendations of level $i-1$ is denoted by $Trust_{A,X,i}$.
  - *Certificates.* $Cert_{X,Y}$ denotes the fact that A holds a certificate for Y's public key issued and signed by entity X.
  - *Recommendations.* $Rec_{X,Y,i}$ denotes the fact, that A holds a recommendation of level i for entity Y issued and signed by entity X.

  The inference rules that allow the derivations of statements from already

known statements are defined by Maurer as follows:

$$Aut_{A,X},\ Trust_{A,X,1},\ Cert_{X,Y} \ \vdash \ Aut_{A,Y}\ , \tag{4.3}$$

$$Aut_{A,X},\ Trust_{A,X,i+1},\ Rec_{X,Y,i} \ \vdash \ Trust_{A,Y,i}\ . \tag{4.4}$$

The statements can thereby be divided into two different categories. The first category gives information about the *characteristic* of nodes and contains $Aut_{A,X}$ and $Cert_{X,Y}$. The second category holds information about the trustworthiness of nodes' *characteristics* and contains the statements $Trust_{A,X,i}$ and $Cert_{X,Y,i}$. Note that all these statements are deterministic, so trust in a node's *characteristic* means total trust. Due to the different levels i of trustworthiness statements, it is possible to infer statement chains of arbitrary length. In Maurer's model the general aim of building those chains is to infer new *Aut* statements. Thus a chain of statements could be built as follows: [91, Example 3.4.]

$$Aut_{A,X}, Trust_{A,X,2}, Rec_{X,Y,1}, Cert_{X,Y}, Cert_{Y,B} \vdash Aut_{A,B}\ ,$$

since:

$$Aut_{A,X},\ Trust_{A,X,2},\ Rec_{X,Y,1} \ \vdash \ Trust_{A,Y,1}\ ,$$
$$Aut_{A,X},\ Trust_{A,X,1},\ Cert_{X,Y} \ \vdash \ Aut_{A,Y}\ ,$$
$$Aut_{A,Y},\ Trust_{A,Y,1},\ Cert_{Y,B} \ \vdash \ Aut_{A,B}\ .$$

The first simplification of [91] yielding a reduction of complexity, especially in the computations of the probabilistic part, is to restrict the trustworthiness statements to level 1, while disallowing the use of second-hand evidence. For the purpose of building a pure trust model, we also redefine Maurer's *statements* as follows:

- *Trust.* $Trust_{X,Y}$ denotes X's belief that a particular entity Y is a trustworthy TA member.
- *Distrust.* $Distrust_{X,Y}$ denotes X's belief that a particular entity X is generally *not* a trustworthy TA member.
- *Authenticity of public keys.* $Aut_{A,X}$ denotes A's belief that a particular public key P_X is authentic.

To incorporate *negative evidence* in a deterministic model, it is necessary to define an additional parameter for distrust. Further *statements* such as *Aut* that might deliver information about a node's trustworthiness can also be defined. Limiting the length of trust chains to 1, inference rules are defined as follows:

$$Trust_{A,X},\ Trust_{X,Y} \vdash Trust_{A,Y}\ , \tag{4.5}$$

$$Trust_{A,X},\ Distrust_{X,Y} \vdash Distrust_{A,Y}\ , \tag{4.6}$$

$$Trust_{A,X},\ Aut_{X,Y} \vdash Aut_{A,Y}\ . \tag{4.7}$$

Rules (4.5) and (4.6) represent the forwarding of trust information over one hop, while (4.7) shows the mechanism to include additional statements in the model.

- **Probabilistic part** The deterministic model part defined all parameters of the trust model such as fixed statements and inference. This allowed the deduction of all implicitly available statements. Our probabilistic augmentation adds the notion of uncertainty to statements in a continuous certainty range $[0, 1]$. Every event is true only with a certain probability, and the core of the probabilistic part is to determine the certainty of the inferred events (statements). The following provides a brief summary of the model; for details on the base model of Maurer we refer to [91].

The set of statements which are contained in a node A's view is denoted by $View_A$. The closure of $View_A$ under the inference rules (4.5)–(4.7) is then labelled with $\overline{View_A}$, and contains the whole statement knowledge of node A, including inferred statements. Since every statement shall be certain in a range from 0 to 1, the certainty of a statement is represented by the probability that this statement is true, and the probability $P(S \in \overline{View_A})$ is labelled as the *confidence value*.

The probability of an inferred statement S from node A is the probability of this statement being inferable from statements included in $View_A$, i.e., $S \in \overline{View_A}$. With $\mathcal{S}_A$ denoting the power set of $View_A$, the confidence value $conf(S)$ for a statement S can be defined as $\text{conf}(S) = P(S \in \overline{View_A}) = \sum_{\mathcal{V} \subseteq \mathcal{S}_A : S \in \overline{\mathcal{V}}} P(\mathcal{V})$.

The model defined so far allows to specify arbitrary dependencies between the statements in $\mathcal{S}_A$. Having limited the level of inferences to 1, $P(\mathcal{V})$ can be computed as:

$$P(\mathcal{V}) = \prod_{S \in \mathcal{V}} p(S) \cdot \prod_{S \notin \mathcal{V}} (1 - p(S)) \ .$$

Finally, the probability $p(S)$ for a derived statement S can be obtained as

$$p(S) = \text{conf}(S) = \sum_{\mathcal{V} \subseteq \mathcal{S}_A : S \in \overline{\mathcal{V}}} \prod_{S \in \mathcal{V}} p(S) \cdot \prod_{S \notin \mathcal{V}} (1 - p(S)) \ ,$$

where the most costly, but due to the limitation to inference level 1 still practical, computable part is the determination of the set $\{\mathcal{V} \subseteq \mathcal{S}_A : S \in \overline{\mathcal{V}}\}$.

A crucial point in every trust system is the initial determination of trust. We assume that trust is anchored either by physical contact (e.g., intervisibility with a compromised node) or by evidence gathered by an IDS during the scenario.

Determining one trust value as input for the cluster metric requires that both the *confidence values* for *Trust* and *Distrust* are combined to one trust factor $1 \geq t_f \in \mathbb{R}$. Every strategy that overstates one of the values would provide a potential point of attack. In the case of a strong effect of the Distrust value for example, an attacker could spread negative evidence about a node's neighbours and thus isolate the node from all its friendly neighbours. In order to minimise the ability of such attacks, we calculate the final trust factor f_t as $f_t = \frac{1}{2} + \frac{(\text{conf}(Trust) - \text{conf}(Distrust))}{2} \in [0, 1]$, where 0 means maximum distrust, 0.5 means no or a neutral opinion, and 1 means maximum trust.

4.4.2.2 Signal strength metric

To avoid a permanent transmission breakdown between a node and it is TA connection, it is desirable to choose a nearby node as TA connection. Since the distance between two nodes does not necessarily represent their connection quality, we choose the signal strength as a measure for the proximity of nodes. The signal strength is commonly specified in dBm, and the benchmark data are provided by the maximal transmission power (100 mW = 20 dBm using the IEEE 802.11 standard as an example) and the threshold for the minimal required receiving power of -80 dBm [4]. Since dBm already provides a logarithmised and thus feasible measure for the original mW values, we use the dBm values to define the signal strength factor f_s for a signal strength of s dBm as $f_s = \frac{s+80}{100}$.

4.4.2.3 Energy metric

Limited battery power is one of the major constraints in mobile ad hoc networks. Since TA nodes generally perform a higher interaction with their neighbours than ordinary nodes, it is desirable to choose TA members with a sufficient battery level.

Most modern battery systems provide a direct or indirect metric based on the voltage of the batteries decreasing with the percentage of discharge $disc \in [0,1] \subseteq \mathbb{R}$, proportional to $1-\sqrt[3]{disc}$ [42]. We therefore define the energy metric as $f_e = 1-\sqrt[3]{disc}$.

4.4.2.4 Routing metric

Although the set of TA nodes which is determined by our cluster algorithm would provide a suitable routing backbone, our approach is also intended to fit into a network with a preselected routing protocol. As stated in Section 4.4.1, a TA overlay network might be bootstrapped without performing additional data transfer. Under the premise of an existing routing protocol, we define the *routing metric* in a way that takes advantage of already established routes. For this purpose we use the number of destination nodes rdn (routing destination nodes) that a node has reached within a certain time period rtp (routing time period), as a measure for its activity in the routing process. The value for rtp needs to be defined depending on the routing protocol. The parameter rpn (routing perfect node) defines a benchmark for the number of reachable destination nodes. A node is of significant importance for the routing process, i.e., part of many routes, if $rdn \geq rpn$. According to this convention, we define the *routing value*, which is the output of the *routing metric* as follows:

$$f_r = \begin{cases} 1 - \frac{rpn-rdn}{rpn} & , \ rdn \leq rpn \\ 1 & , \ rdn > rpn \ . \end{cases}$$

However, if there is no predefined routing protocol and the routing is performed using the TA nodes as backbone, this metric is not required and hence not included in the cluster metric.

4.4.2.5 Bandwidth metric

While the routing metric encourages the concentration of data transfer to a small number of nodes, this can exceed the nodes' bandwidth. To avoid delays or dropped packets, the *bandwidth metric* measures the load of a node with respect to its available bandwidth. As before, we use the IEEE 802.11g standard for our example. In 802.11g the data rate at a certain point in time is dependent on the signal strength, and varies between 8 values from 6 Mbps[3] to 54 Mbps. We label these values dr_1 (data rate 1) to dr_8, where dr_1 represents the lowest rate of 6 Mbps and dr_8 the highest rate of 54 Mbps, respectively. Moreover, we define $\mathrm{dr}_c = \lfloor dr \rfloor$ as the highest dr_i that is lower than dr, and $d_i,\ 1 \leq i \leq 8$ denotes the respective data rate. We assume that a data rate at least *two* levels above the minimum required level is sufficient to achieve the desired throughput dr. Based on this assumption, we define the *bandwidth factor* f_b as follows:

$$f_b = \begin{cases} 0 & , \ \mathrm{dr}_c \geq \mathrm{dr}_i \\ \frac{dr - \mathrm{dr}_c}{2 \cdot (\mathrm{dr}_{c+1} - \mathrm{dr}_c)} & , \ \mathrm{dr}_{i-1} \leq \mathrm{dr}_c < \mathrm{dr}_i \\ 0.5 + \frac{dr - \mathrm{dr}_c}{2 \cdot (\mathrm{dr}_{c+1} - \mathrm{dr}_c)} & , \ \mathrm{dr}_{i-2} \leq \mathrm{dr}_c < \mathrm{dr}_{i-1} \\ 1 & , \ \mathrm{dr}_c < \mathrm{dr}_{i-2} \ . \end{cases}$$

4.4.2.6 Cluster metric

All component metrics were designed to firstly return a value in $[0, 1]$ to provide a linear correlation between their return value and its relative importance. We chose a linear correlation as it allows us to define several partial metrics in an intuitive way, and additionally allows a straight forward definition of the cluster metric. For example, a battery level of 50 % is considered as "half as good" as a totally charged

[3] 1 Mbps = 10^6 bits per second

battery, and a signal of 100 % (20 dBm) is twice as good as a signal of 50 % -30 dBm). According to this simplified approach, the cluster metric finally combines all partial metrics in a linear combination and returns the quality factor in Definition 1. Let $\mathbb{M} = \{t, s, e, r, b\}$ be the set of indices of all partial metrics, and f_t, f_s, f_e, f_r, f_b be the respective return values based on the information of a node i about a node j at a certain time. Then the quality factor r_{ij} is calculated as:

$$r_{ij} = \begin{cases} \sum_{i \in \mathbb{M}} \lambda_i \cdot f_i & (f_i \geq 0 \ \forall i \in \mathbb{M}) \\ 0 & \text{otherwise} , \end{cases}$$

with:

$$\sum_{i \in \mathbb{M}} \lambda_i = 1, \quad \lambda_i \geq 0 \ \forall i \in \mathbb{M} .$$

The configuration of the λ values is discussed in the following Section 4.5.

4.5 Evaluation and analysis

4.5.1 Simulations

For evaluating our cluster algorithm and for investigating configurations with different values of the loading factors λ_i of the cluster metric, we have implemented the proposed model in the network simulator NS-2. We set up three different simulation scenarios: Simulation 1 is a random waypoint scenario investigating the influence of the cluster metric, Simulation 2 and 3 examine the stability and efficiency of the cluster algorithm based on scenarios defined in Section 3.2.

Simulation 1: Influence of the cluster metric The purpose of Simulation 1 is to investigate the influence of the metrics from Section 4.4 in the choice of

the TA nodes. We chose random node mobility for this simulation to examine the development of trust values in a highly dynamic network with frequent network topology changes. We ran a simulation containing 48 benign and 2 hostile nodes. In our simulations, benign nodes were able to find out about a node's affiliation if the distance between the nodes was 10 m or less. After 100 seconds in a 700 m $\times$ 700 m area, all friendly nodes had a trust value about the two enemy nodes of 0.1 or smaller, and thus did not choose them as TA nodes at all.

In a second simulation, nodes were only configured to attack for a time period of 30 seconds, such that only two other nodes had physical contact with these hostile nodes during these 30 seconds. The purpose of this scenario was to examine the effect of Byzantine behaviour on our algorithm. Even after 15 minutes, the other nodes were changing their opinions about the two temporarily hostile nodes. This shows the difficulty of providing security against temporarily misbehaving nodes (as might be the case for Byzantine nodes), and the need for intrusion detection systems that quickly detect nodes' misbehaviour.

Further simulations for the same setup were performed to illustrate the quantitative effect of different configurations for our cluster metric. Figure 4.1 shows four

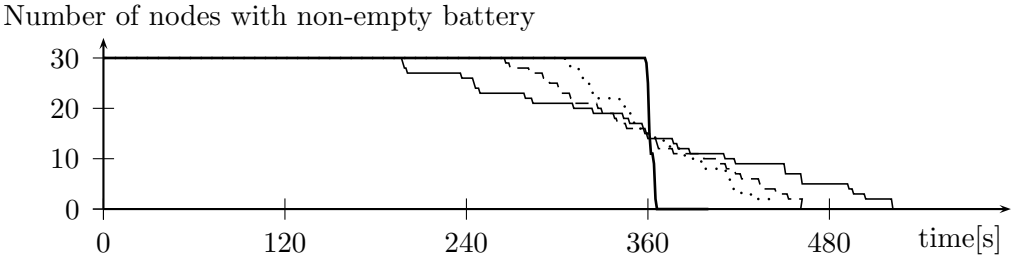


Figure 4.1: Number of nodes with sufficient battery level.

different configurations of the node id and the battery level, while the other metrics were not considered, i.e., loaded as 0. The left graph (thin solid line) presents the lifetime of the nodes for the loading (node id, battery level) = $(1, 0)$, which corresponds in the case of a 1-hop cluster to Amis' original cluster algorithm [3]. For this con-

figuration, the nodes start running out of energy after 200 seconds. The other three graphs display the lifetime of the nodes for the configurations $(2/3, 1/3)$ (dashed line), $(1/3, 2/3)$ (dotted line) and $(0, 1)$ (thick solid line), and thus the influence of our energy metric. In the configuration $(0, 1)$, all nodes live as long as possible and run out of energy at almost the same time after 360 seconds. These results show the impact of our quality factor and its scalability with respect to incorporating our energy metric.

Due to constant changes in the energy level as well as other parameters which have an impact on our cluster metric, the quality value of the nodes are changing permanently. To avoid the frequent swapping of the TA nodes, we introduced Extension 2 to our cluster algorithm (see Section 4.4). We use the simulation Scenarios 1 and 2 as defined in Section 3.2 to examine the stability of the cluster algorithm and the robustness against node failures.

4.5.2 Stability and reliability of the distributed TA

In Section 4.4 we discussed the changes to our cluster-based algorithm for TA member selection. In this section we illustrate, based on two simulation scenarios, the behaviour of our cluster-based selection algorithm with respect to three aspects:

- total number of TA nodes (cluster heads);
- number of nodes successfully connected to the TA;
- number of received packets/second.

The first aspect describes the number of TA nodes at every time interval in our simulation scenarios, and shows the influence of formation changes, interaction of radio waves with the topography and the transmission power on the total amount

of TA nodes. Since our TA member selection algorithm is intended to perform the basis for a security architecture in which several secret shares are distributed among the TA members, the number of TA nodes is a decisive factor.

The second aspect (number of nodes that are successfully connected to the TA) reveals how many nodes in the network have chosen a node as TA member (cluster head) that is indeed capable of acting as a TA node. We define a node to be successfully connected to the TA if its TA node is indeed a member of the TA, and a physical connection is still existent. The continuous exchange of cluster packets would yield a perfectly informed network and thus enable every node to immediately react to connection breakdowns and changes in the behaviour of neighboring nodes. However, since transmission is a crucial factor for the energy consumption in MANETs, we aim to maximise the interval between cluster messages, while keeping the connectivity to the TA nodes at a sufficient level.

The third aspect (number of received packets) illustrates the additional data overhead caused by the cluster algorithm. Since the amount of transmitted packets per second in our model can simply be calculated as "the number of nodes in the network divided by the frequency of cluster messages", we examine the number of received packets as a metric for the data overhead.

Simulation 2: Cluster algorithm behaviour in a city area Simulation 2 models a platoon of soldiers traversing a city area as defined in Scenario 2 in Section 3.2 and illustrated in Figure 3.7. The group of 37 nodes is moving between buildings, splitting up in three subgroups and merging again. This simulation is intended to expose the effect of abrupt communication breakdowns as well as the division of the network in several subgroups. Figure 4.2 through Figure 4.4 show the behaviour of the network with respect to aspects 1–3 for different frequencies of the cluster message exchange. We ran our simulations for frequencies of multiples

of 2 seconds up to 16 seconds, and chose 2, 4 and 8 seconds to give a clear overview of the results.

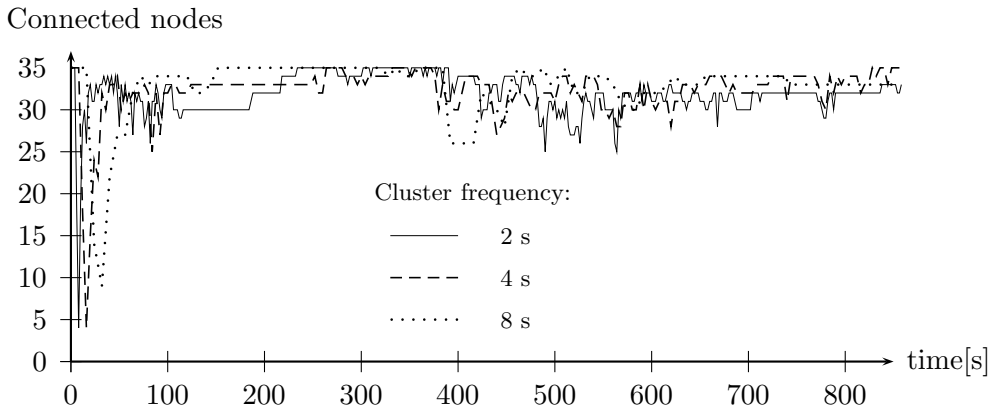


Figure 4.2: Simulation 2: Number of nodes connected to the TA.

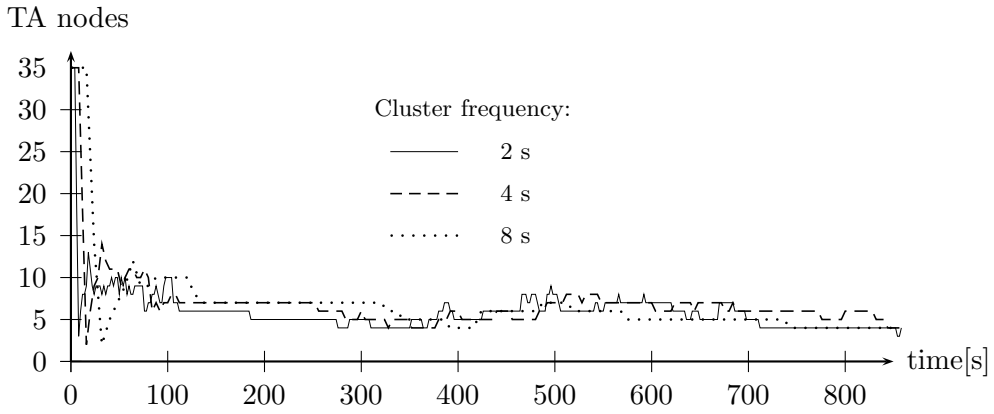


Figure 4.3: Simulation 2: Number of TA nodes.

We set the initialisation time (see Section 4.4) to 100 seconds. In this time the nodes are configured to choose their best TA connection independently from already established TA nodes. After the initialisation time, the reelection of TA members is encouraged to avoid frequent changes of TA nodes (see Section 4.4). Figure 4.2 shows that directly after the start, only 5 to 10 nodes are successfully connected to a TA node, while this number increases to more than 30 nodes in the course of time. The duration of this configuration process is dependent on the frequency of cluster messages. In the case of a cluster message frequency of 2 seconds (solid line) this process lasts only 20 seconds, while it takes 80 seconds in the case of

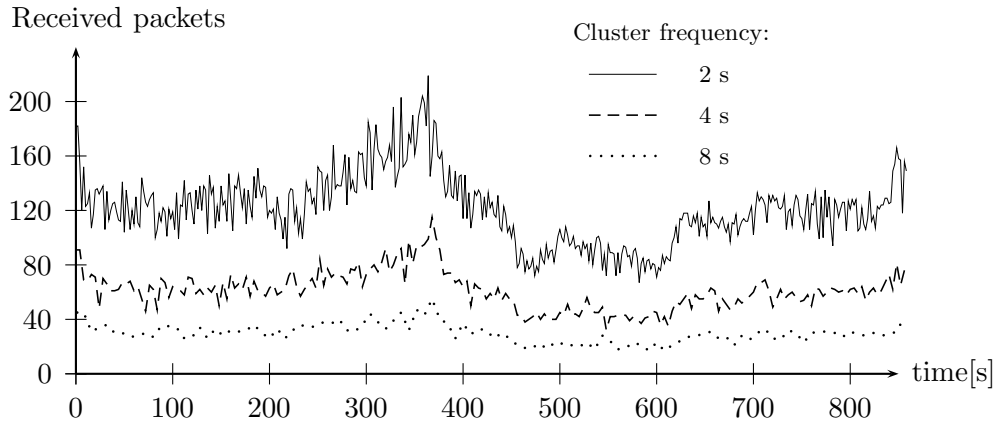


Figure 4.4: Simulation 2: Total number of received cluster packets per second.

a cluster frequency of 8 seconds (dotted line). Accordingly, our cluster algorithm needs approximately 10 rounds of cluster message exchanges to shape a sufficient set of TA nodes.

After approximately 380 seconds (see Figure 3.7(a) in Section 3.2), the nodes start to split up between the buildings. The effect on the connectivity of the network and the TA can be seen in Figure 4.2 and 4.4. The number of received packets increases until second 380, since the platoon needs to choose a closer formation to move in between the buildings. Thereupon this number decreases abruptly due to communication breakdowns. As an impact on the connectivity to the TA in case of a cluster frequency of 8 seconds (dotted line), up to 9 of the 35 nodes temporarily lose their connection to the TA. In case of a cluster frequency of 2 (solid line) or 4 (dashed line) seconds, the algorithm reacts more quickly, and only 5 nodes lose their connection to the TA.

A further crucial observation are the fluctuations in Figure 4.2 between second 400 and 700, especially for a cluster frequency of 2 seconds (solid line). This behaviour occurs when the nodes from different small groups get a temporary connection between buildings, as can be identified in Figure 3.7(b). This problem does

not occur for a cluster frequency of 8 seconds (dotted line), since the nodes of different small groups will not receive enough cluster messages to choose the new node from another group as TA member.

Despite the fact that the number of received packets increases after the collation of the group (Figure 3.7(c)), this event has no notable effect on the TA or the connectivity of the network. As a further important result of Figure 4.2 through Figure 4.4, the cluster algorithm shows a very similar behaviour for the different cluster message frequencies of 2, 4 and 8 seconds. In view of bootstrapping a security architecture on top of the TA, the consequence of this observation is that even in a network with numerous abrupt communication breakdowns, a cluster frequency of 8, or possibly more seconds, is still sufficient.

Simulation 3: Influence of unreliable nodes and links Simulation 3 models different movement techniques of a platoon of soldiers in a hostile area, as defined in Scenario 1 in Section 3.2 (Figure 3.6). Figure 4.5 through Figure 4.7 show the behaviour of the network with respect to aspects 1–3 during the simulation for different cluster message frequencies of 4, 8 and 16 seconds. We changed the frequencies to 4, 8 and 16 seconds for this simulation, since longer durations between the exchange of cluster messages are desireable and Simulation 2 already showed that cluster frequencies of 4 and 8 seconds are feasible. Additional simulations, based on a message frequency of 8 seconds and illustrating the influence of node failure, loose contacts of the wireless devices and different amounts of transmission power, are shown in Figure 4.8 through Figure 4.10.

The behaviour during the initialisation time of 100 seconds is similar to Simulation 2, where approximately 10 rounds of cluster message exchange are required to first shape a sufficient set of TA nodes (Figure 4.5). Subsequently, the number of received packets decreases due to the formation stretching of the travelling platoon

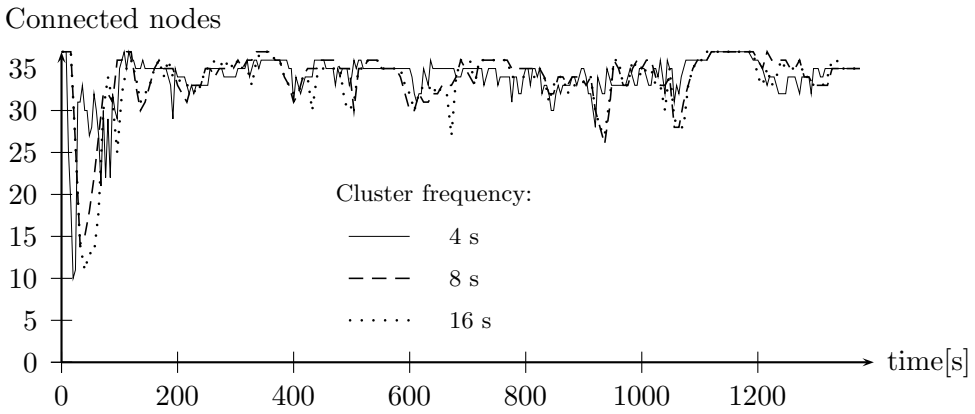


Figure 4.5: Simulation 3: Number of nodes connected to the TA.

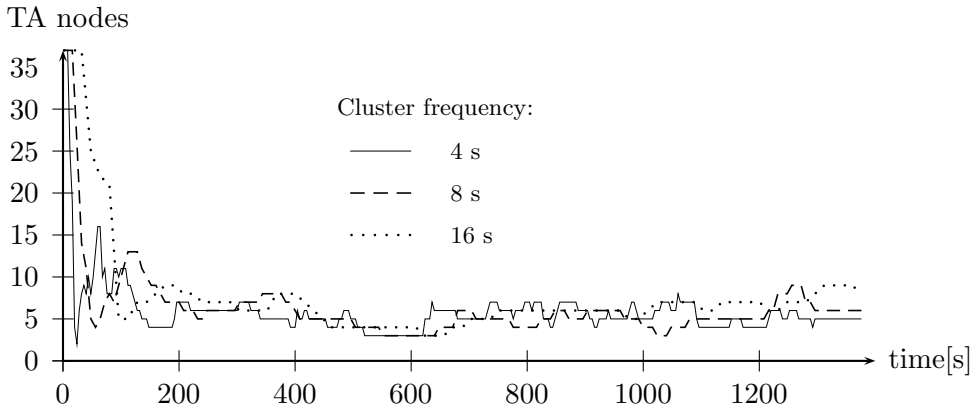


Figure 4.6: Simulation 3: Number of TA nodes.

during second 150 and 320 (Figure 3.6(a)). The partition into three squads which move in a compact formation until second 600, increases the number of received packets, while the following formation change to a "stretched line" (Figure 3.6(b)) abruptly decreases this number. Nevertheless, these changes in the connectivity of the network have almost no effect on the connectivity of the TA (Figure 4.5) and the number of TA nodes (Figure 4.6).

The only two noticeable events in the rest of the simulation that come with a short decrease of the connectivity to the TA, are the resumption of speed after the file formation in Figure 3.6(b) and the division of the network in second 1070

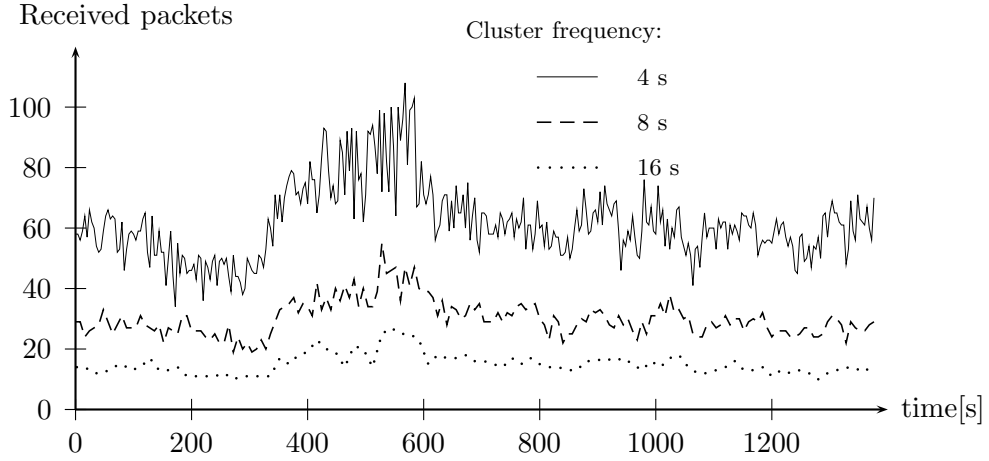


Figure 4.7: Simulation 3: Total number of received cluster packets per second.

(Figure 3.6(c)) while crossing the danger area. In summary, our algorithm for the distribution of the TA works smoothly and does not have any weak points in this simulation scenario. For further refinement of the algorithm we have examined the influence of node failures, loose contacts and different amounts of transmission power, as illustrated in Figure 4.8 through Figure 4.10. The wireless devices of the nodes in the first of these scenarios begin to drop out during the enemy contact from second 700 to 800 (Figure 3.6(b)). There are only slight differences between the network containing 20 failing nodes (dashed line) and the network without node failures (dashed line in Figure 4.8). The first apparent influence can be observed in the case of 25 failing nodes, as illustrated by the dotted line.

We performed the same simulation as illustrated in Figure 4.8 a second time with loose contacts of the wireless devices instead of node failures. Loose contacts were simulated by a random failure in sending and receiving packets of 50 %. Even 25 failing nodes only has minor impact on the connectivity to the TA, while a loose contact of 20 devices has no visibly negative effect.

Finally we also ran Simulation 3 under different transmission strengths of 1 mW, 5 mW and 15 mW, yielding a communication range in free space of approximately

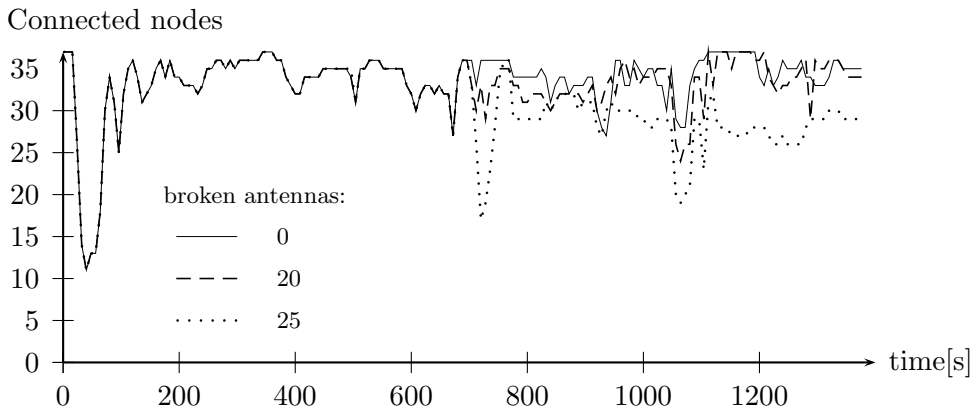


Figure 4.8: Simulation 3: Influence of the breakdown of several nodes. Cluster frequency: $8s$.

30 m, 45 m and 60 m, respectively. Our algorithm appeared to be sensitive to the stretching of the formation in case of a communication range of 60 m (Figure 4.10 solid line). In the period of 150 to 320 seconds, as well as after the division of the network after 600 seconds, the increasing distances between the nodes disconnected up to 15 nodes from the TA. This can be traced back to the higher communication range, which enables the nodes to connect to distant TA members that move out of the connected node range at the aforementioned events. In networks with high communication ranges, the *quality value* of our cluster algorithm should therefore be configured in a way that encourages the choice of nearby nodes, i.e., by using the signal strength metric.

4.6 Summary

In this chapter we have investigated cluster algorithms for establishing a distributed trust authority in MANETs. We modified an existing cluster algorithm to:

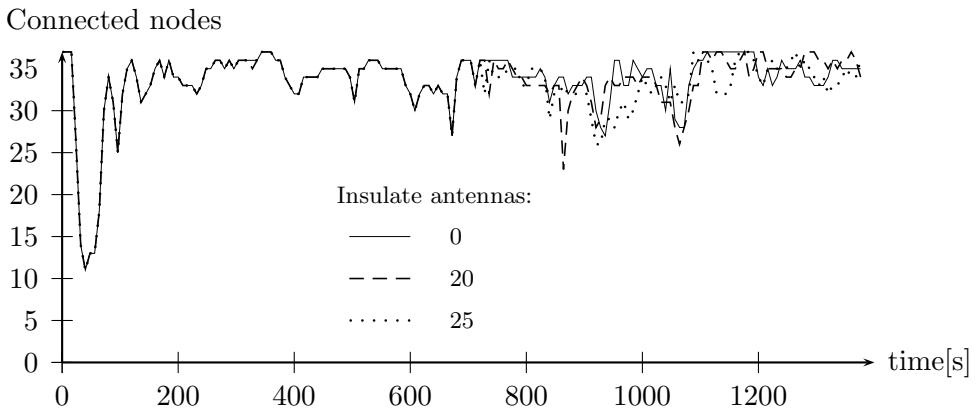


Figure 4.9: Simulation 3: Influence of loose contacts of several nodes
Cluster frequency: $8s$.

- incorporate a trust metric to provide robustness against malicious nodes, relying on the accuracy of the underlying intrusion detection system;
- be configurable by several metrics to provide a tradeoff between efficiency, reliability and security;
- require less communication overhead;
- change cluster heads withs a low frequency, making the cluster heads a feasible set of nodes for a distributed trust authority.

Simulations have shown the robustness of our developed cluster algorithm under node failures and mobility, as well as significant gains in efficiency while adding the ability to incorporate higher-layer properties such as trust. Our analysis has shown that cluster algorithms allow a distributed trust authority to be determined in MANETs. This TA can dynamically react to network changes and can help to allow scalability of the energy level of the network and other parameters such as trust. The incorporation of a trust metric in the cluster head selection mechanism constrains the election of suspicious nodes as TA members. An open research problem is

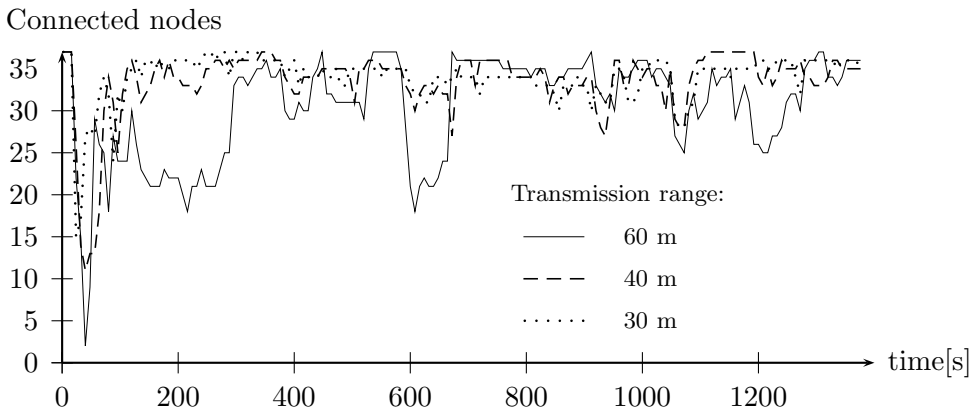


Figure 4.10: Simulation 3: Influence of different amounts of transmission power. Cluster frequency: 8s.

the development of adversarial models for cluster algorithms and other network layer protocols, which allow to quantify the robustness against malicious nodes. In Chapter 8 we discuss possible directions for the development of such adversarial models.

Chapter 5

Distributing symmetric keys

Contents

In this chapter we propose two protocols for non-interactive key agreement. We prove the resilience of both schemes against a large number of malicious nodes and

investigate their feasibility for MANETs regarding computational and communicational costs.

5.1 Introduction

Key agreement is a fundamental tool for secure communication; it lets two nodes in a network agree on a shared key that is known only to them, thus allowing them to use that key for secure communication (see Section 2.3.3).

In environments where bandwidth is at a premium, there is a significant advantage to *non-interactive* schemes, where two nodes can compute their shared key without any interaction. The classical Diffie-Hellman key agreement protocol [45] is an example of a non-interactive scheme: in that protocol, node A can compute a shared key with node B knowing only the public key of B (and its own secret key). However, the nodes in this protocol must still learn each other's public keys somehow, which implies either direct communication between them or some other form of coordination.

To minimise the required coordination, one may use *identity-based* key agreement, where the public key of a node could be the node's identifier. Such schemes rely on a central authority with a master secret key that provides each node with a secret key that corresponds to that node's identifier. In this setting, the non-interactive identity-based scheme of Sakai *et al.* [123] allows node A to compute a shared key with node B knowing only B's identity (and A's own secret key).

However, in MANETs it is often unrealistic to expect all nodes to register with just one central authority as required by Sakai *et al.* [123]. One would therefore prefer a *hierarchical* system, where a root authority only needs to distribute keys to

a small number of large organisations, and each of these can further distribute keys to smaller and smaller units, until finally the end-nodes get their secret keys from their immediate organisational unit.

In this chapter we propose two schemes that have all the above functional properties and are secure in a strong sense. That is, they are *non-interactive* to save on bandwidth, *identity-based* to save on coordination and support ad hoc communication, and *hierarchical* to allow for flexible provisioning of nodes. At the same time, we design these schemes to be *resilient* to the compromise of *any number of end-users (leaf nodes)* and resilient to the compromise of a threshold number of nodes in the upper levels of the hierarchy.

One of our proposed schemes is computationally very efficient but requires larger keys, making it especially suitable for MANETs where keys are distributed during pre-configuration which will be used during the life-time of the MANET. The second scheme is computationally slightly less efficient but has very small key sizes, making it attractive for MANETs where online key refreshing is required. We run simulations to investigate the online distribution of the key material using the second scheme. This work was accomplished in close collaboration with IBM Research.

5.2 Background

In the context of symmetric key agreement protocols in Section 2.3.3, we already discussed the works of Sakai *et al.* [123], Blundo *et al.* [22], and Eschenauer and Gligor [50] (and its extension by Ramkumar *et al.* [109]), which play a central role in our construction.

There are also a few prior attempts to improve the resilience of the scheme of

Blundo *et al.*. Hanaoka *et al.* [63] show that in a sparse system (where most pairs of nodes never need to communicate) the threshold can be increased by a significant factor (possibly up to 16 fold) without adversely effecting the performance. That solution is applicable in relatively static networks where one can partition the nodes into disjoint sets and have no inter-set communication, but it is not applicable in settings where every pair of nodes may potentially need to communicate.

Another technique for improving the resilience of the Blundo *et al.* scheme was proposed by Zhang *et al.* [152], using random perturbations in order to randomise the polynomials used in the protocol of Blundo *et al.*. However, a practical instantiation of the parameters for the protocol enables the parties to agree on a small number of bits (say 12) in each execution of the protocol. Thus in order to generate enough secret keying material, about ten independent executions of the protocol need to be carried out. Furthermore, this scheme does not provide the hierarchical capabilities.

Matt [89] described some trade-offs between resilience and performance, and even proposed a combination of the schemes of Blundo *et al.* and Sakai *et al.* that is similar to ours. However, his scheme requires that each node *communicates directly with the central authority*, and hence it is not a hierarchical scheme.

Following the identity-based encryption scheme of Boneh and Franklin [24], Horwitz and Lynn [68] initiated a study of hierarchical identity-based encryption. Interestingly, their scheme combines a pairing-based scheme and a polynomial-based one, as we do. However, they only use two levels, where the pairing-based scheme is placed at the top level and the polynomial-based scheme at the second level. In this work we reverse the order, using the polynomial-scheme for all the top levels and the pairing-based scheme only for the leaves, to obtain a solution that supports non-interactive key agreement.

5.3 Preliminaries

Our key agreement schemes (KAS) are built by combining the identity-based key agreement protocol of Sakai *et al.* [123] with hierarchical schemes that use linear operations, such as the polynomial-based key distribution system of Blundo *et al.* [22] or the random-subset-based scheme. Below we present some background material and recall these schemes.

5.3.1 Bilinear maps and the BDDH assumption

Let G_1 and G_2 be two cyclic groups of order q for some large prime q. Let e be a mapping $e : G_1 \times G_1 \to G_2$. The mapping e is:

1. Bilinear if $e(P^a, Q^b) = e(P, Q)^{ab}$ for any $P, Q \in G_1$, $a, b \in Z_q$.
2. Non-degenerate if e does not send all pairs to the identity in G_2.
3. Computable if there is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Bilinear mappings that can be computed efficiently are known based on Weil and Tate pairings in Abelian varieties.

Bilinear Decisional Diffie-Hellman Problem (BDDH).
The central hardness assumption on which we base our schemes is the following BDDH assumption introduced by Boneh and Franklin [24]. Let G_1, G_2 and e be as above. Given a random $P \in G_1$, $P^a, P^b, P^c \in G_1$ for random $a, b, c \in Z_q$, and given $h \in G_2$, it is hard to distinguish the case where $h = e(P, P)^{abc}$ from the case where $h = e(P, P)^r$ for a random and independent $r \in Z_q$. Formally, an algorithm $\mathcal{A}$ has

advantage ϵ in solving the BDDH in $\langle G_1, G_2, e\rangle$ if

$$Pr[\mathcal{A}(P, P^a, P^b, P^c, e(P,P)^{abc}) = 1] - Pr[\mathcal{A}(P, P^a, P^b, P^c, e(P,P)^r) = 1] \geq \epsilon,$$

where the probability is over the random choice of $P \in G_1$, $a, b, c, r \in Z_q$, and the internal randomness of $\mathcal{A}$. The BDDH assumption (with respect to $\langle G_1, G_2, e\rangle$) states that feasible adversaries can have only an insignificant advantage. In this chapter we forgo the asymptotic notation that is needed to make this formal. Instead we take the "concrete security" approach, directly relating the advantage of an adversary against our scheme to the advantage in solving BDDH over the relevant group.

5.3.2 Non-interactive identity-based key agreement

Sakai *et al.* [123] propose the following non-interactive (but not hierarchical) key agreement scheme. The central authority sets up the parameters for an identity-based public key system, by fixing two cyclic groups G_1, G_2 and the bilinear map $e : G_1 \times G_1 \rightarrow G_2$. Furthermore, it chooses a cryptographic hash function $H : \{0,1\}^* \rightarrow G_1$. It then chooses a secret key $s \in Z_q$ and provides a node with identity ID with the secret key $S_{ID} = H(ID)^s \in G_1$.

The shared key between two nodes ID_1 and ID_2 is $K = e(H(ID_1), H(ID_2))^s \in G_2$, which party ID_1 computes as $K = e(S_{ID_1}, H(ID_2))$ and ID_2 computes as $K = e(H(ID_1), S_{ID_2})$.

The security of this scheme can be reduced to the BDDH assumption in the random-oracle model, as was shown in [49].

5.3.3 Polynomial-based KAS

Our generic KAS presented in Section 5.4 can be instantiated using different hierarchical systems. Here and in the next subsection we describe two instantiations of such hierarchical systems. The first is based on multivariate polynomials and follows Blundo *et al.* [22] (we refer to it as Blundo's scheme). Let L be the depth of the hierarchy, i.e., the nodes are arranged in a tree with L levels. Each node's identity corresponds to the path from the root to the node (thus a node at level i will have as identity a vector with i components $\langle I_1, \ldots, I_i \rangle$ where each I_j is an integer).

For desired threshold parameters $\{t_i : i \leq L\}$, the root authority chooses a random polynomial (over Z_q for a large enough prime q) $F(x_1, y_1, \cdots, x_L, y_L)$, where the degree of x_i, y_i is t_i. F is chosen such that $F(x_1, y_1, \cdots, x_L, y_L) \equiv F(y_1, x_1, \cdots, y_L, x_L)$, i.e., F is symmetric between the x_i and y_i. One way to choose such polynomial is to choose a random polynomial f on the same variables, and then set $F(x_1, y_1, \cdots, x_L, y_L) = f(x_1, y_1, \cdots, x_L, y_L) + f(y_1, x_1, \cdots, y_L, x_L)$. We note that the size of the description of F (number of coefficients) is $\Pi_{i=1}^{L} \frac{(t_i+1)(t_i+2)}{2}$, so this scheme can only be used with moderate thresholds t_i and small values of L.

The master secret of the system is the polynomial F. The secret key of the node with identity I in the first level of the hierarchy is the polynomial $F_I = F(I, y_1, x_2, y_2, \cdots)$ that has $2L - 1$ variables. Similarly, the secret key of a node at level i with identity $\vec{I} = \langle I_1, \ldots, I_i \rangle$ is the polynomial $F_{\vec{I}} = F(I_1, y_1, \cdots, I_i, y_i, x_{i+1}, y_{i+1}, \ldots)$ that has $2L - i$ variables, and the secret key of the leaf with identity $\langle I_1, \ldots, I_L \rangle$ is the polynomial in L variables $F(I_1, y_1, \cdots, I_L, y_L)$.

The shared key between the two leaf nodes $\langle I_1, \ldots, I_L \rangle$ and $\langle J_1, \ldots, J_L \rangle$ is the value of the polynomial $F(I_1, J_1, \ldots, I_L, J_L) = F(J_1, I_1, \ldots, J_L, I_L)$, that each node can compute by evaluating its secret polynomial on the points that correspond to

its peer's identity.

Blundo's secret sharing scheme provides information theoretic security for uncompromised nodes in the following important way. We call a node *compromised* if the attacker has learned *all* of the node's secrets (i.e., all the coefficients of the polynomial the node holds, and hence all of its descendants' shared keys), otherwise we call it *uncompromised*. Blundo's scheme guarantees that the key shared between any two uncompromised nodes is information theoretically secure, namely, all values of the key are equally possibly given the attacker's view.

Note that a node N in the hierarchy can be compromised (i.e., all its secrets learned) by directly breaking into N and finding its secrets or by breaking into other nodes from which the information in N can be reconstructed. For example, one can learn all of N secrets by breaking into an ancestor of N or by breaking into $t+1$ of its children (where t is the node's threshold). Here, the word "secrets" can refer to the coefficients of the polynomial held by a node N or, equivalently, to the set of pairwise shared-keys known to N and its descendants (i.e., the set of keys shared by these nodes with every other node in the hierarchy). In general, since pairwise keys are derived by evaluating a polynomial, the knowledge of a set of secrets (coefficients and/or pairwise keys) can allow an attacker to derive the value of additional secrets. Given a set of secrets S, we say that a key K (e.g., between parties I and J) is *independent from* S if no attacker (even if computationally unbounded) can learn anything about K from the set S; we say that a set of keys S is independent if each key in it is independent of the other keys in the set. It can be shown that in a Blundo's hierarchy with $L+1$ levels (with the root being at level 0 and the leaves at level L) and threshold t_i at level i, an attacker that wants to learn all the secrets of a node N in level ℓ must learn (at least) a set of T independent keys where T equals

$$\Pi_{i=\ell+1}^{L} \frac{(t_i+1)(t_i+2)}{2} \Pi_{i=1}^{\ell}(t_i+1).$$

In particular, the attacker must learn *at least* this many number of keys (or coefficients) in the system before it can learn all of N secrets.[1]

5.3.4 Subset-based KAS

A different instantiation of our KAS uses subset-based key pre-distribution schemes, which were first studied by Eschenauer and Gligor [50]. In such schemes the root authority chooses a large number of secret keys for its key-ring, the key-ring of every node contains a random subset of these keys, and the shared key for two nodes is computed from the intersection of the keys in their respective key rings.

Extending it to a hierarchical ID-based scheme is fairly straightforward: a parent node in the tree gives to each child a random subset of its key ring, and that subset is computed deterministically from the child's name (using a cryptographic hash function). Such a hierarchical scheme was described by Ramkumar *et al.* [109].

The scheme works as follows:

[1] When all t_i are equal to the same number t we have $T = (\frac{(t+1)(t+2)}{2})^{L-\ell}(t+1)^{\ell}$.

- The parameters of the system are the number of keys at the root (denoted N), and for each level i in the tree a probability $p_i \in (0,1)$, which says what fraction of the key ring of the parent is forwarded to the children.

- The root node chooses N secret keys at random for its key ring. For our purposes, we think of these keys as integers modulo a fixed large prime number q.

- Let $n = \langle I_1, \ldots, I_i \rangle$ be a node at level i with key ring $R_n = \{K_1, K_2, \ldots\}$, and let $c = \langle I_1, \ldots, I_i, I_{i+1} \rangle$ be a child of n in the tree. The node n uses a cryptographic hash function to derive a sequence of numbers from the child's name, say by computing: $r_j \leftarrow H(c, j)$. The child c gets all the keys $K_j \in R_n$ for which $r_j < p_i$. Namely, its key ring is $R_c = \{K_j \in R_c \ : \ r_j < p_i\}$.

- For two leaf nodes $\langle I_1, \ldots, I_L \rangle$ and $\langle J_1, \ldots, J_L \rangle$, the nodes repeat the hash calculations from above to determine the intersection of their key rings, and the shared key is computed as the sum modulo q of all the keys in the intersection.

It is not hard to show that in order to withstand up to t_i compromised nodes at level i, the optimal setting for the parameter p_i is $p_i = 1/(t_i + 1)$. And given all the t_i and p_i, the parameter N should be set large enough to ensure the required level of security. Specifically, to ensure that an attacker that compromises up to t_i nodes in each level i will not have more than e^{-m} probability of learning the shared key between two specific uncompromised nodes, the parameter N should be set to $N = \lceil m / \prod_i p_i^2 (1-p_i)^{t_i} \rceil \approx m e^L \cdot \prod_i t_i(t_i+1)$. To ensure that the attacker will have probability at most e^{-m} to learn the key of *any* pair of uncompromised nodes, we need to add to the number N above $2 \log M$ where M is the number of nodes in the system.

5.4 Our fully leaf-resilient KAS

Our goal is to provide a hierarchical identity-based key agreement scheme that is secure against compromise of any number of nodes at the lowest level of the hierarchy. Namely, we consider a KAS in the form of a tree-like hierarchy of authorities that issue keys to nodes lower in the hierarchy, where any two leaf nodes can compute *without interaction* a shared key unique to these two leaves. That is, each leaf computes the shared key from its own secret key, its peer's identity, and potentially some other public information.

We want this hierarchy to be secure in the sense that an attacker that compromises some of the nodes in the hierarchy cannot learn the keys shared by leaves that are not in the subtree of a compromised nodes. Typically, the above guarantee of security will only hold as long as the attacker does not compromise too many nodes, and we extend this guarantee even in the face of an unlimited number of compromised leaves.

Technically, our scheme is a combination of *linear* hierarchical schemes (of which the schemes from Sections 5.3.3 and 5.3.4 are special cases) and the identity-based scheme of Sakai *et al.* that was described in Section 5.3.2. In the rest of this section we formalise the linear requirement from the underlying hierarchical KAS and then present our hybrid scheme.

Definition 4 (Linear Hierarchical KAS) *A hierarchical key agreement scheme is called* linear *if it satisfies the following properties with respect to some linear space V and an integer parameter N: (i) The root authority selects N random elements from V to be used as the* master secret keys*. (ii) The secret key of each node in the hierarchy consists of a set of values $v_1, v_2, \ldots \in V$, each of which is a* linear combination *(over V) of the master secret keys. (iii) The shared key between every two nodes is an element of V which is also a linear combination over V of the master*

secret keys. (iv) The number of values v_i in each node and the coefficients in the linear combinations that determine these values are derived deterministically *from public information such as the position of a node in the hierarchy and its identity.*

We note that in typical hierarchical schemes, an internal node will provide its children with values that are linear combination of its own values (which thus must be linear combinations of the master secret keys). This is indeed the case for the two schemes from Sections 5.3.3 and 5.3.4.

5.4.1 A leaf-resilient hybrid hierarchical KAS

We now show how to combine a linear hierarchical KAS $\mathcal{H}$ with the bilinear identity-based scheme of [123] (see Section 5.3.2), resulting in a hybrid scheme, $\mathcal{H}'$, that is as resilient to attack on the *internal* nodes as $\mathcal{H}$ is, but which is *fully resilient* against leaf compromise. Roughly, a leaf node with identity ID can compute the shared key "in the exponent", thereby obtaining the secret $H(ID)^s$ as needed for the scheme of Sakai *et al.*.

Our scheme can be described as follows. Let $\mathcal{H}$ be an L-level linear hierarchical KAS, and we construct an $L+1$-level hybrid KAS $\mathcal{H}'$ as follows:

- The root authority of $\mathcal{H}'$ sets up and publishes the parameters for an identity-based public key system, by fixing two cyclic groups G_1, G_2 of order q and the bilinear map $e : G_1 \times G_1 \to G_2$, as well as a hash function $H : \{0,1\}^* \to G_1$.

  In addition, the root authority carries the same actions as the root authority of $\mathcal{H}$, where the linear space over which $\mathcal{H}$ is defined is set to Z_q.

- For any internal node other than the root, a leaf or a parent of a leaf, all actions are identical to the scheme $\mathcal{H}$.

- A node F that is a parent of a leaf has secret values $v_1, \ldots, v_n \in Z_q$ as in $\mathcal{H}$. For each child leaf ℓ with identity[2] ID_ℓ, the values that F provides to ℓ are the elements $H(ID_\ell)^{v_i} \in G_1$, $i = 1, \ldots, n$.

- The shared key between leaf nodes ℓ, ℓ' with identities ID, ID' whose parents are F, F', respectively, is computed as follows:

  Let $v_1, \ldots, v_n$ be the secret key of F, and let $\alpha_1, \ldots, \alpha_n$ be the coefficients of the linear combination that F would have used in $\mathcal{H}$ to compute a shared key with F'. In other words, F would compute the shared key with F' in $\mathcal{H}$ as $s = \sum_i \alpha_i v_i \pmod q$. Recall that the secret key of ℓ are the group elements $V_1 = H(ID)^{v_1}, \ldots, V_n = H(ID)^{v_n} \in G_1$, and that the coefficients α_i can be computed from publicly available information. The leaf ℓ computes

  $$U_1 \leftarrow \prod_i V_i^{\alpha_i} \quad \left(= H(ID)^{\sum_i \alpha_i v_i} = H(ID)^s\right)$$

  and $U_2 \leftarrow H(ID')$, and sets the key to $K \leftarrow e(U_1, U_2) = e(H(ID), H(ID'))^s$. Similarly the leaf ℓ' with secret key $V'_1, \ldots, V'_{n'}$ determines the coefficients $\beta_1, \ldots, \beta_{n'}$ that F' would have used in $\mathcal{H}$, then computes $U'_1 \leftarrow H(ID)$ and $U'_2 \leftarrow \prod_i (V'_i)^{\beta_i}$ and sets $K \leftarrow e(U'_1, U'_2) = e(H(ID), H(ID'))^s$.

For example, when applying this hybrid to the subset scheme from Section 5.3.4, the two leaves will determine the set of indexes I for which they both received keys, and then the leaf ℓ will compute $U_1 \leftarrow \prod_{i \in I} V_i$ and the leaf ℓ' will compute $U'_2 \leftarrow \prod_{i \in I} V'_i$.

Security A rigorous analysis and proof of the above generic hybrid scheme is presented in our paper [57].

[2]We assume that the identity includes the entire path from the root of the hierarchy to the leaf, so no two leaves have the same identity.

5.5 Implementation and simulations

There are many trade-offs that one can make when choosing a key agreement scheme for a particular application. Below we describe some of these trade-offs:

5.5.1 Setting the thresholds

The complexity of the schemes that we present here is proportional to the product $\prod_i t_i$, so to get a realistic scheme one must choose the t_i's as small as the security considerations allow. As was explained in the introduction, if the hierarchy is expected to only have a very small branching factor (except for the leaves), then one can set the t_i's to this expected branching factor. Otherwise, it might be the case that higher-level nodes are better protected than lower-level nodes, and thus the thresholds t_i should increase as we go down the tree.

Below we demonstrate the complexity that we get for two settings, both of which correspond to a hierarchy that has two levels of intermediate nodes (i.e., the leaves are three levels below the root). The first setting is applicable to a very small tree, where we set $t_1 = t_2 = 3$. The second setting is applicable to a large tree, where we use $t_1 = 7$ and $t_2 = 31$. The resulting key sizes and number of operations to compute the shared key are summarised in Table 5.1.

5.5.2 Polynomials versus subsets

The two underlying hierarchical schemes from Sections 5.3.3 and 5.3.4 offer quite different characteristics. The main advantage of the polynomial scheme is that the secret keys are small: for the same setting of the thresholds, the polynomial scheme has the leafs holding keys of size $\prod_i (t_i + 1)$ group elements, and the root holding a

key that is a square of that, namely $\prod_i ((t_i+1)/2)^2$. The factor of $\frac{1}{2}$ is because the polynomial is symmetric. In the subset scheme, on the other hand, the size of the keys is larger by roughly a factor of me^L (for security level of e^{-m}). In our examples with $L = 2$, and assuming $m = 20$ (which seems to be a reasonable value), the keys in the subset scheme are larger by about two orders of magnitude.

However, computing the shared key between two leaves may be faster using the subset construction. This is because in the polynomial scheme the leaves have to do one elliptic-curve multiplication for every group element in their key, whereas in the subset scheme they only need to do an elliptic-curve addition for every element in the intersection of the two sets (which is a small fraction of the entire key of each of them).

Another difference is the security behaviour: the polynomial scheme ensures security as long as the adversary does not exceed the threshold of nodes compromised, but can break completely once the threshold is exceeded. The subset construction, on the other hand, provides a gradual degradation of security, with the probability of a break monotonically increasing as the adversary compromises more nodes.

Finally, we comment that one can also use hybrids between the two schemes, such as using the subset construction on one level and the polynomial construction

Table 5.1: Performance characteristics of hierarchical schemes.
Subset numbers are listed with respect to security level $e^{-20} \approx 2 \times 10^{-9}$. (add. and mult. stand for "additions" and "multiplications", respectively)

Scheme:	*Polynomial scheme*		*Subset scheme*	
Thresholds:	$t_1 = t_2 = 3$	$t_1 = 7, t_2 = 31$	$t_1 = t_2 = 3$	$t_1 = 7, t_2 = 31$
Key size (# of elements)	Root: 100 Leaves: 16	Root: 19008 Leaves: 256	Root: 28768 Leaves: 1800	Root: 8930800 Leaves: 35000
Shared key Computation	1 pairing 16 EC mult.	1 pairing 256 EC mult.	1 pairing 450 EC add. 1800 hashing	1 pairing 1100 EC add. 35000 hashing

on the other. Such hybrids are discussed in the works of Du *et al.* [48] and Liu and Ning [83].

5.5.3 Concrete implementations

Combining the numbers from Tables 5.1 and 5.2 (and assuming that the SHA256 hashing operation takes about one microsecond on a 2.4 GHz Pentium-4 platform), the running times and storage requirements for the various schemes are summarised in Table 5.3. As is evident by these tables, the polynomial scheme offers much smaller keys, while the subset construction is faster for the leaves (but slower for the parents of the leaves).

In addition to the operations listed in Table 5.3, one also needs to implement the key generation by the root and key delegation between internal nodes. At key generation time the root needs to choose random numbers between 1 and q, one for every group element, where the prime number q is the order of the elliptic curve over which this scheme is implemented. For the curves that we deal with, the prime q is in the range from $q \approx 2^{160}$ to $q \approx 2^{300}$. For the polynomial scheme the time and space requirements are insignificant, and even for the subset scheme this is manageable. At worse, with the parameters $t_1 = 7, t_2 = 31$ and working over a large curve, the

Table 5.2: Elliptic-curve parameters from [97].
Security level is the approximate equivalence in RSA security. $SS(n,-)$ is the curve $Y^2 = X^3 - X - 1$ over $GF(3^n)$. Running times are in milliseconds on a 2.4 GHz Pentium 4. Addition time is an estimate based on the timing of multiplication. Element-size is the number of bits representing a point on the curve.

Security	EC	Addition	Multipl.	Pairing	El.-size
RSA-912	SS(163,-)	0.1 ms	15 ms	57 ms	260
RSA-1080	SS(193,-)	0.12 ms	22 ms	86 ms	307
RSA-1976	SS(353,-)	0.3 ms	94 ms	355 ms	561

Table 5.3: Timing/storage of hierarchical schemes.
The numbers were computed from Tables 5.1 and 5.2, assuming 1μs for computing SHA256.

Scheme:	*Polynomial scheme*		*Subset scheme*	
Security level:	$t_1 = t_2 = 3$ RSA-912	$t_1 = 7, t_2 = 31$ RSA-1976	$t_1 = t_2 = 3$ RSA-912	$t_1 = 7, t_2 = 31$ RSA-1976
Key size	Root: 2000 Byte Leaf: 520 Byte	Root: 713K Byte Leaf: 17952 Byte	Root: 575K Byte Leaf: 58500 Byte	Root: 327 MByte Leaf: 2.34 MByte
Key delegation to leaves	0.24 sec	24.1 sec	27 sec	3290 sec
Shared key computation	0.3 sec	24.4 sec	0.1 sec	0.7 sec

root needs to generate 327 MByte of random data.

Delegating between intermediate nodes in the subset scheme consists only of hashing (in order to determine which keys to delegate). With the parameter $t_1 = 7$, the root needs to do one hash calculation for approximately every 85 numbers in its key ring (since we only need three bits per number in order to select it with probability 1/8, and one application of SHA256 yields 256 bits). Hence the root needs to perform only about 100000 hashing operations, which can be completed in approximately 0.1 seconds. The intermediate nodes need to do even less work to compute the key delegation. However, the keys in the subset scheme are large, so key delegation may take considerable bandwidth.

Delegating between intermediate nodes in the polynomial scheme requires the evaluation of polynomials modulo q. Specifically, every element that a node at level i delegates to a child is obtained as the result of evaluating a polynomial of degree t_i modulo q, which means performing t_i modular multiplications. Since we work with small t_i's and moderate values of q, and since the secret keys in the polynomial schemes consists of at most a few thousand elements, then this is a rather short calculation. In our more computationally-demanding example, with parameters $t_1 = 7, t_2 = 31$, the root needs to evaluate 8192 polynomials of degree

seven (for a total of about 60000 multiplications modulo a 160-bit to 300-bit prime). Extrapolating from reported speeds of modular exponentiations, this can be done in well under one second. For example, the implementation of DSA in `openssl` was reported to perform one 160-bit exponentiation modulo a 512-bit prime in 0.8 millisecond on a 2.4 GHz Pentium-4. Hence a multiplication modulo a 300-bit prime should take no more than 2–3 microseconds, and 60000 of them can be done in under 0.2 seconds. We complement our discussion of implementation issues with a report on a specific simulation scenario in the following section.

5.5.4 Simulation of key distribution

Although the main advantage of a non-interactive key agreement scheme lies in applications where the distribution of keys to the leaves can be done in an offline manner, there are still many applications where one needs to refresh the keys "in the field". In this section we examine the feasibility of our key distribution scheme in such an environment. We build our simulation on the polynomial-based scheme, which is due to smaller key sizes particularly suitable for an online post-deployment key distribution. Specifically, we consider a small tactical network, where complete key refreshing may be mandatory, e.g., to merge networks with different security parameters.

To get a feel for the time and feasibility for an ad hoc key refreshment in a tactical network, we set up a simulation scenario illustrating a platoon of 37 nodes in a city area. For this simulation we chose to work with the setting of $t_1 = 3$, and to use small parameters also for the elliptic curve. We use the Scenario 3 from Section 3.2. Figure 5.1 shows a snapshot of the simulation. Figure 5.1 shows a snapshot of the scenario.

In the simulation scenario, the platoon is splitting between buildings in two,

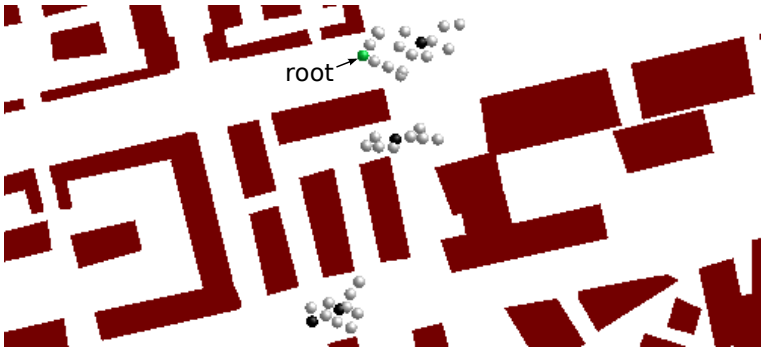
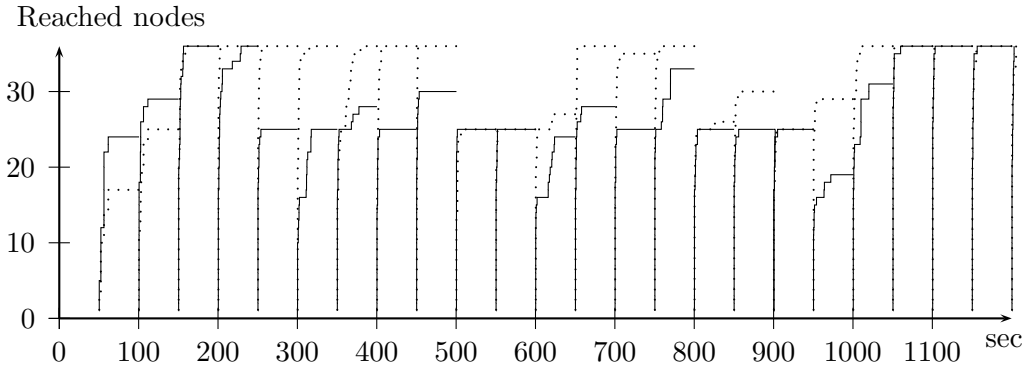


Figure 5.1: Snapshot of our simulation scenario illustrating a platoon. The platoon consists of 37 nodes traversing a city area.

and temporarily three, subgroups, which are represented by squads. Two of the squads remain connected nearly all time, while the connection to the third squad is disrupted several times due to buildings and distances between the squads. The platoon is intended to be part of a bigger network, e.g., a network of several platoons. The node labelled as "root" represents the root (level 0) node, the black nodes are level 1 nodes and the gray nodes are leaf nodes in the hierarchy of depth 2. The simulation examines a key distribution process, which is executed with a frequency of 50 seconds. For this purpose, the root node disseminates the key material to all level 1 and leaf nodes. As part of a bigger network, the root node can either calculate the respective keys on its own, or receive it from a key distribution centre. The simulation starts at the point where the root node already holds the key material and simulates its distribution in time steps of 50 seconds. This is not supposed to be a realistic frequency for key refreshing, rather we want to investigate several times whether the keys can be successfully distributed. The simulation does not incorporate response messages from internal and leaf nodes or retries in case of transmission failures.

The keys for level-one nodes in this scheme consist of $4^2 = 16$ numbers modulo $q \approx 2^{160}$, and the size of leaf nodes consist of 16 points on the elliptic curve. Hence the size of the key material packets for level-one nodes is $16 \times 160 = 2560$ bit (320



Reached:	29	36	25	25	25	24	25	25	25	19	36	36
Duration:	11.6	29.0	17.5	3.5	1.6	24.0	2.5	4.0	2.0	22.1	3.5	2.9
Data Packets (size including header: 383 to 583 byte)												
Sent:	38	79	37	47	26	31	31	48	31	28	48	65
Received:	32	58	31	30	25	26	26	35	27	21	38	55
Dropped:	6	16	6	17	1	4	5	12	4	4	10	8
AODV Packets (average size including header: 87 byte)												
Sent:	79	149	69	75	85	51	55	76	50	47	102	97
Received:	32	66	31	31	25	26	26	35	27	22	38	55
Dropped:	33	47	29	38	46	19	24	36	19	18	48	25

Figure 5.2: Simulation results of the key distribution in an interval of 50 sec. The tables beneath show the results for every 50 seconds of the distribution process. solid line: buildings interrupt communication as shown in Figure 5.1; dashed line: buildings are ignored.

bytes), and the size of the key material packets for leaf nodes is $16 \times 260 = 4160$ bits (520 bytes)[3].

The graph in Figure 5.2 shows the number of nodes that received their new key at various points in time. New keys are distributed at a frequency of every 50 seconds. The dotted line illustrates the results using the standard Two Ray Ground propagation model as incorporated in NS-2 [51], which does not consider buildings. The figure shows that the distance between the nodes in the periods of 150 to 450, 650 to 800 and 1000 to 1200 seconds is small enough to reach all nodes in the network. The solid line displays the results under the consideration of buildings

[3]To transmit the key material securely, it can be encrypted by the old private key as proposed by Balfe *et al.* [9].

as obstacles and reflections on house walls up to a depth of 2 by our ray-optical propagation model (see Section 3.1.2). In the period of 250 to 1000 seconds, the root node mostly only reaches 25 nodes, which is the size of the upper two squads in the simulation. Despite several moments of intervisibility between the squads, the complete key material cannot be reliably distributed. In the time periods from second 350–370 and 750–770, for instance, several connections between the squads exist, but the routing protocol seems to react too slowly to take full advantage of these temporarily existing routes. These results indicate that we cannot expect a successful key distribution in a weakly connected network similar to the illustrated one in second 250 to 1050.

Beneath the graph in Figure 5.2, the first row lists the maximum number of nodes that could be reached after the respective time. Values are listed for multiples of 100 seconds for the simulation that incorporates influences by buildings (solid line). The duration values show that the routing process keeps sending packets up to a period of 30 seconds. If the links in the network are stable, the distribution process is completed after 1.5 to 5 seconds. This situation occurs in second 150 and 1000 to 1200, when all nodes have a line-of-sight contact, as well as in seconds 400, 500 to 550 and 800 to 900, where the upper two squads are properly connected but the third one is disconnected from these two due to buildings or distance. The values for the data and routing packets give an overview of the amount of data that is sent during a key distribution. The high amount of dropped routing packets highlights the potential for sophisticated communication strategies, which will be investigated in future work. However, distribution times of 1.5 to 5 seconds at times when the network is well connected, and the computation period of less than one second, show that an ad hoc key refreshment in our hierarchical key generation scheme is feasible for hierarchy of depth 2 (not counting the root node).

5.5.5 Summary

The bottleneck of our scheme appeared to be the generation of the leaves' secret keys. While the calculation of the master key and the internal nodes' key material takes only 0.76 seconds, the leaves' key generation takes approximately 10 seconds and turns into the major part of the calculation. To keep the overall execution period at a feasible level, the calculation of the keys needs to be distributed among the internal nodes. As stated in the examination above, the key of every node can be determined by all nodes superior in the node's hierarchy-branch up to the KDC. According to our NS-2simulation, the calculation of the leaves' key polynomials could be distributed among the 5 superior nodes, where each of these nodes would calculate at most 6 leave keys. Thus the computation period would be 0.76 seconds in the first instance for creating the master secret and the key for the internal nodes, and $6 \cdot 0.32 = 1.92$ seconds in the second instance, yielding a total period of 2.68 seconds. Since the dissemination of the key material is split into two parts, where in each part the respective nodes are at a 1-hop distance, we expect the distribution period to remain at approximately 3 seconds. The combination of the key generation and the key distribution therefore yields a total time period of approximately 5 seconds.

5.6 Summary

In this chapter we have proposed, and analysed, hierarchical non-interactive key agreement protocols which are particularly suitable for use in MANETs. The emphasis of our schemes is on being resilient to compromises of arbitrary numbers of leaf nodes (which are considered the most vulnerable). While our schemes are limited in their efficiency as the thresholds grow, this is not an impediment for networks with the number of nodes and limited hierarchies typically found, for example, in tactical networks. The proposed schemes are intended to minimise the communica-

tion complexity both by small key sizes and a decentralised key distribution due to the hierarchical structure. Simulation of the key distribution process demonstrates that an online key refreshing requires 1.5 to 5 seconds if the nodes in the MANET are connected.

Part II

Secure distributed protocols in MANETs

Chapter 6

Reliable execution of security protocols

Contents

In this chapter we present a novel algorithm for enhancing the efficiency and robustness of distributed trust authority protocols for MANETs. Our algorithm selects a set of TA nodes that are best suited to perform a distributed computation such as a threshold signature using a suite of metrics for measuring the efficiency and reliability of candidate nodes.

6.1 Introduction

To avoid a single point of vulnerability or failure, many TA security services in MANETs are distributed using (k, n)-threshold secret sharing schemes (see Section 2.4). A node wishing to obtain certification of a public key, refresh a private key in an identity-based public key infrastructure or revoke a key using quorum-based decision making, must typically interact with at least $k + 1$ TA nodes to successfully complete a protocol.

In contacting these $k + 1$ TA nodes, the current MANET literature largely assumes that all TA nodes are equally viable as service providers. However, in all distributed security architectures (see Section 2.4), nodes can either assign themselves as TA members or are selected as TA members based on the network topology. In Chapter 4 we developed a cluster algorithm to select the TA member nodes based on parameters such as trust and battery level, to provide a more elaborated choice of the distributed TA. To contact TA nodes, a requesting node typically floods the network with service request messages in the hope that they contact the necessary number of TA nodes. Unfortunately, the level of interactivity required to support such schemes may become problematic in ad hoc networks due to the limited energy capacity of nodes, as well as bandwidth constraints on the communication links between nodes. Excessive amounts of inter-node communication can quickly deplete a node's energy reserves, as well as potentially clogging the channel over which multiple nodes must communicate, ultimately disrupting the provision of security services within the network.

In this chapter we take the view that TA nodes are not equal, and that it should be possible to specify which individual TA nodes participate in a protocol request. For instance, a group of TA nodes may be deemed to be more reliable, better connected, have greater energy reserves, be within a certain geographically

bounded area or be considered more trustworthy than another group of TA nodes. To this end, we present a novel route optimisation algorithm that determines a suitable set of TA nodes (as judged against a specified set of criteria metrics), and that provides a routing path to contact these nodes with minimum communication overhead. Our algorithm further balances the need for resource efficiency with the ability to reliably complete a distributed TA service within a bounded time-frame (with a specified success probability). If our algorithm finds a satisfactory solution, it returns a set of TA nodes and an appropriate (Pareto-optimal) routing strategy for contacting them. If no initial solution can be found matching our constraints, our algorithm returns a set of suitable TA nodes and a calculated success probability to reach $k+1$ of these nodes (based on average number of prior successful interactions). In the latter case, the returned nodes can either be contacted directly using unicast protocol, or, alternatively, the success probability, time frame and/or criteria metric parameters of our algorithm can be relaxed for a revised run of our algorithm in an attempt to find a more efficient routing solution.

6.2 Background

There has been significant research on distributed security protocols for ad hoc networks in recent years [153, 78, 87, 147]. Beginning with the work of Zhou and Haas [153], much research has been carried out on distributing traditionally centralised *Certification Authority* (CA) functionality over multiple nodes within a network [78, 87]. In [153], nodes are pre-designated as either CA or non-CA nodes. A non-CA node wishing to obtain certification must contact at least k-out-of-n CA nodes via a reliable broadcast channel. The use of network broadcast techniques as a means of reliably contacting TA nodes has also been studied in [147]. To reduce the high communication overhead generated by flooding, later proposals have suggested the use of β-unicast as a mechanism for contacting TA nodes [148]. This mechanism

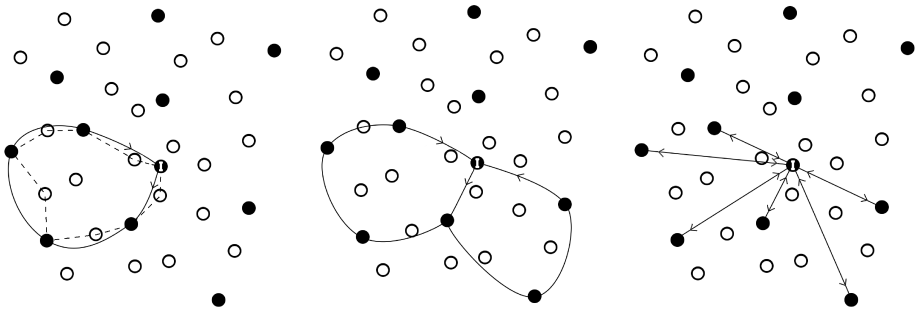
relies on multiple individual unicasts targeted on individual TA nodes. Carter *et al.* [31] achieve a further reduction of the communication overhead by investigating *manycast* routing protocols for contacting TA nodes. However, missing from the majority of these proposals is the criteria leading to the selection and use of TA nodes. With the exception of [148], many authors leave the selection of TA nodes to some unexplained and unexamined policy layer. In [148], the authors suggest that TA nodes should be elected based upon battery level, transmission range and physical protection, but their proposed algorithms for contacting TA nodes ignores this information in composing a suitable routing strategy.

6.3 Communication algorithm

In this section we outline our route optimisation algorithm for contacting selected TA nodes. Our algorithm assumes an overlay network of TA nodes that operates on top of a physical MANET. A requesting non-TA node initially contacts his nearest TA node, who in return contacts k additional TA nodes (using an efficient routing strategy). To select these additional TA nodes, we assume the initiating TA node has the following information: the location of TA nodes in its neighbourhood, e.g., all TA nodes that are at most c physical hops away; the length of routes (number of hops) between these TA nodes; and any pertinent properties of these TA nodes, e.g., energy-reserves, degree of physical protection etc. We assume that this information is publicly available (or at least derivable) from network observations.

Our algorithm's goal is to find an overlay route containing a desired set of TA nodes. We leave the routing between the non-TA nodes to the underlying MANET routing protocol. We denote a non-splitting route, which is starting and ending at the same TA node, as a *single loop* path. The *length* of a single loop path is the number of TA nodes it contains, excluding the point of contact TA node.

Figure 6.1(a) shows an example of a single loop of length 4. We refer to a loop consisting of one or more single loops as a *splitting loop*, and the individual single loops within this splitting loop as *partial loops*, see Figure 6.1(b). The length of a splitting loop is the length of the longest partial loop it contains.



(a) Single loop.

(b) Splitting loop with two partial loops.

(c) Splitting loop with six partial loops.

Figure 6.1: Single and splitting loops.
TA Node = Black, Non-TA Node = White, TA Overlay Route = Solid Line, Physical Route = Dashed Line.

We model the communication cost (or *cost*) of either a single or partial loop by the number of transmissions required to complete the loop, approximated by the number of physical hops. For example, the route in Figure 6.1(a) contains the initially contacted TA node, 4 other TA nodes, and 5 non-TA nodes. The number of hops, and thus the communication cost of the route shown in Figure 6.1(a) is therefore 10. Our routing optimisation strategy, as shown in Figure 6.1(a) and 6.1(b), applies only to distributed protocols in which the payload does not dramatically increase as the route is traversed. For protocols unsuitable for this sort of execution, every node must be contacted independently. However, we note that this strategy can be expressed using multiple splitting loops. Figure 6.1(c) shows a splitting loop in which every TA node is independently contacted.

6.3.1 Probability for success and expectation values

Once the initiator TA node has begun a distributed protocol, several factors can impact the protocol run. In particular, nodes along the path may fail or refuse to participate in the protocol (either as result of temporary overloading or because of Byzantine behaviour). Consequently, one or more partial loops may fail, either through non-participation of one or more TA nodes or through the (partial) collapse of the path due to networking issues. In either case, less than the desired k TA nodes will have been contacted and the protocol may have to be (partially) restarted.

In order to estimate the probability for the occurrence of partial failure, every TA node stores the results from previous protocol runs and sets the probability of each event (non-participation and path collapse) to the average outcome of prior interactions. We denote p_j as the probability that a contacted TA node n_j will successfully contribute to the protocol, and $p^{(i)}$ as the probability that a single loop (or partial loop) of length i will not collapse. The values of p_j and $p^{(i)}$ are based on potentially incomplete information, and cannot incorporate all (statistical) dependencies between the effects of non-participation and loop collapse. Consequently, p_j and $p^{(i)}$ can only provide approximative probability values. We initially set both probabilities to 0.5.

Based on p_j and $p^{(i)}$, we calculate probabilities and expectation values for the communication cost, and for the number of TA nodes that will successfully contribute to the protocol run. The probability of successfully reaching k nodes in a splitting loop L is given by:

$$P^{(L)}(k) = \sum_{(J,\mathcal{I})\in\mathcal{K}_L^*} \left(\prod_{j\in J} p_j \cdot \prod_{j\in L\setminus J} (1-p_j) \cdot P_{\mathcal{I}} \right) ,$$

where

$$P_{\mathcal{I}} = \sum_{I \in \mathcal{I}} \left(\prod_{i \in I} p^{(i)} \cdot \prod_{i \in \mathcal{I}_L \setminus I} (1 - p^{(i)}) \right) \ .$$

$\mathcal{K}_L^*$ contains all possible combinations of TA nodes and non-collapsing loops in which exactly k TA nodes are successfully contacted. Thus, $\mathcal{K}_L^*$ consists of tuples $(J, \mathcal{I})$ which define a set of TA nodes in J, and the corresponding sets of non-collapsing single loops of length i in I. Furthermore, $\mathcal{I}_L$ is the set of all single loops of length i contained in L. Consequently, the expectation value $E_N(L)$ for the number of nodes that can be reached in a splitting loop L can be calculated as:

$$E_N(L) = \sum_k P^{(L)}(k) \cdot k \ . \tag{6.1}$$

If an insufficient number of TA nodes have been reached for the distributed computation, the initiator TA node may need to contact the remaining TA nodes in one or more additional *rounds*. Let $L^1_{k_m}$ denote the splitting loop that is executed in round 1, and $L^m_{k_m}$, $m \geq 2$, denote m alternative loops to be used in the following rounds. Furthermore, let $L^m_{k_m}$ denote the splitting loop in a possible configuration

$$conf := \left\{ L^1_{k_1},\ (L^2_{k_2})_{1 \leq k_2 \leq k_1}, (L^3_{k_3})_{1 \leq k_3 \leq k_2},\ \ldots \right\} \ ,$$

that is executed in round m, if a number k_m of TA nodes remains to be contacted after $m-1$ rounds. The probability $P(S,k)$, that the initial k TA nodes can be contacted after $\hat{m}$ rounds is given by:

$$P(S,k) = \sum_{\tilde{m}=1}^{\hat{m}} \sum_{\vec{k} \in K_{\tilde{m}}} \left(\prod_{m=1}^{\tilde{m}} P^{(L^m_{k_m})}(k_{m+1}) \right) \ , \tag{6.2}$$

where $K_{\tilde{m}}$ is given by:

$$K_{\tilde{m}} = \{(k_1, \ldots, k_{\tilde{m}+1}) | k_1 = k, k_{\tilde{m}+1} = 0, k_i \geq k_{i+1}\} \ .$$

Accordingly, the expectation value $E_C(S)$ for the communication cost of a configuration $conf$, under the assumption that k nodes could be reached, can be calculated as:

$$E_C(S) = \sum_{\tilde{m}=1}^{\hat{m}} \sum_{\vec{k} \in K} \left(\prod_{m=1}^{\tilde{m}} P^{(L^m_{k_m})}(k_{m+1}) \cdot Cost(L^m_{k_m}) \right) . \tag{6.3}$$

6.3.2 Greedy communication algorithm

Our Greedy algorithm consists of three stages. In the first stage, our algorithm chooses an initial set of TA nodes and a routing strategy that contacts each of these nodes independently (Figure 6.1(c)). In the second stage, our algorithm successively searches for a more efficient routing strategy using splitting loops, which always contain a subset from the TA nodes determined in the first stage. The third stage of our algorithm simply compares all routing strategies from the first and the second stage, and returns the most optimal (as judged against our fitness criteria of timeliness, probability of success and the quality metrics of the TAs) as the algorithm's output.

It is possible that the algorithm executes several iterations during one distributed protocol computation. If less than the desired k nodes were reached in the first (or a subsequent rounds), the remaining TA nodes can be used to complete the distributed protocol in future rounds. We denote the TA nodes that have not been contacted in a previous round as G.

First stage: The first stage chooses the sets $g_1, g_2 \subset G, g_1 \cap g_2 = \emptyset$ of TA nodes for the current protocol run. TA nodes are added to sets g_1 and g_2 as follows:

I: Our algorithm first determines the expected number of nodes that can be con-

Algorithm 6.1: Communication algorithm pseudocode.

```
   Input: minProb, k, quality metrics, maxT, set of all nodes
   /* First stage:  Initialize the set of possible configurations C          */
 1 P = 0; /* P : Probability to reach k nodes by current configuration conf  */
 2 while P < (1 + minProb)/2 do
 3 |   if E_N(L^{g_1}) ≤ E_N(L^{g_2}) then
 4 |   |   add new node to g_1 (based on quality metrics);
 5 |   else
 6 |   |   add new node to g_2 (based on quality metrics) ;
 7 |   end
 8 |   determine direct way to reach nodes in g_1 and g_2 (current configuration conf);
 9 |   calculate new P value for conf;
10 |   if  P ≥ minProb then
11 |   |   add conf to C;
12 |   end
13 end
14 if  C = ∅ then
15 |   return conf;
16 end
   /* Second stage:  Find loops to contact subsets of g_1 and g_2            */
17 while P ≥ minProb do
18 |   find loop to current choice of g_1 and g_2;
19 |   calculate new P value for conf;
20 |   if  P ≥ minProb then
21 |   |   add conf to C;
22 |   |   (Step A)
23 |   end
24 |   remove last added node from g_1 or g_2;
25 |   (Step B)
26 end
   /* Third stage:  Return best configuration                                */
27 minCost = infinity;
28 bestConf = NULL;
29 for conf in C do
30 |   calculate new P value for conf;
31 |   while P ≥ minProb do
32 |   |   remove node from g_2;
33 |   |   calculate new P value for new conf;
34 |   end
35 |   if cost for conf < minCost then
36 |   |   minCost = cost for conf;
37 |   |   bestConf = conf;
38 |   end
39 end
40 return bestConf;
```

tacted directly from either g_1 or g_2. To balance the chance of contacting the same number of nodes from g_1 and g_2, a new node will be selected for inclusion in g_1 if $E_N(L^{g_1}) \leq E_N(L^{g_2})$ (see Equation 6.1) and for g_2 otherwise, see step II.

II: In this step, our algorithm searches for a suitable node to add to either g_1 or g_2, based upon the nodes perceived quality metrics. We use a quality value, as used before for our cluster algorithm in Chapter 4, to consider all the desired quality metrics for a given node. This quality value r_j of a node n_j is a result from several metrics f_i, and outputs a value between 0 and 1.

Let $\mathbb{M}$ be the set of desirable properties (e.g., battery level, trust, etc.), then the quality value is calculated as: $r_j = \sum_{i \in \mathbb{M}} \lambda_i f_i$, where λ_i is the weighting factor for property i (adding up to 1): $\sum_{i \in \mathbb{M}} \lambda_i = 1, \lambda_i \geq 0$. In order to factor in a node's connectivity to the network, we define a new *connectivity metric* f_c with the function $d(n_j, n_k)$ calculating the number of hops (distance) between node n_j and n_k.

$$f_c(n_j) = \min \left\{ \frac{\sum_{n_k \in g_1} n_k}{k \cdot d(n_j, n_k)} - \frac{\sum_{n_k \in g_2} n_k}{k \cdot d(n_j, n_k)} + \frac{\sum_{n_k \in G \setminus (g_1 \cup g_2)} n_k}{2k \cdot d(n_j, n_k)}, 1 \right\}$$

The purpose of the connectivity metric is to positively consider the topological proximity of the TA node n_j to all g_1-nodes, and negatively consider the proximity to g_2-nodes. TA nodes that do not belong to either g_1 or g_2 are also considered positively, but only with half weighting[1]. This allows us to obtain two sets, g_1 and g_1, where the nodes in each set can reach each other either directly or over a few hops.

For example, let us assume that connectivity and battery level are the dominant factors in determining node placement in g_1. The connectivity is measured by f_c and the battery level is measured by the energy metric f_e, as defined in

[1]If a node is to be added to the set g_2, then g_1 and g_2 need to be reversed in the equation.

[117]. Then the quality value $r_j = \frac{1}{2}f_c(n_j) + \frac{1}{2}f_e(n_j)$ is calculated for every node $n_j \in G$. The node with the highest quality value will be added to g_1.

III: This process ends when the probability $P(S,k)$ has reached $\frac{1+minProb}{2}$, or when no more TA nodes are available, i.e., $G = g_1 \cup g_2$.

The set $g_1, |g_1| \geq k$ will now contain the nodes that the initiating TA will try to contact, in order to perform a distributed computation. If the distributed computation can be completed successfully by contacting TA nodes from g_1, the set g_2 will not be used. However, if some of the nodes in g_1 fail to participate in the distributed protocol, then the distributed computation will fail. If $k_1 < k$ TA nodes could be reached in the first attempt, then $k_2 = k - k_1$ TA nodes remain to be contacted. To contact these remaining nodes, nodes from g_2 can be used.

The algorithm terminates here if $P(S,k) < minProb$, i.e., no configuration could be found for the given constraints. Otherwise, the algorithm proceeds with the second stage to find more efficient routing strategies.

Second stage: The second stage consists of two iteratively executed steps, A and B (see Algorithm 6.1). Step A determines an alternative loop to contact the current nodes in g_1. Step B removes a single node from g_1. The second stage begins with a single execution of step A in order to find an alternative route for the nodes in g_1. Then steps A and B are applied consecutively until the probability to reach k TA nodes is less than $minProb$. Step B next simply removes one node from either g_1 or g_2; nodes are removed in the opposite order in which they were added in the first stage. The heuristic strategy to determine the alternative (improved) route in step B merges successively those single loops to the current configuration which yield the highest E_N value. Thus, this stage begins with a single loop L with the highest $E_N(L)$ value. As long as nodes remain in g_1 (which are not included in L), a further

loop is added to L, i.e., combined with L to form a splitting loop.

Third stage: The third stage compares all of the configurations that have resulted from the first and second stages, and chooses the configuration with the smallest cost expectation value $E_C(S)$ as the final result. The TA then uses the selected nodes to attempt its computation with the appropriate routing strategy.

Table 6.1: Average simulation results from 50–150 nodes.

simulation area $[m]$	400 × 500					
number of nodes	50			75		
k	2	3	4	2	3	4
Costs (direct)	16.06	25.57	35.99	15.19	23.73	33.25
Costs (proposed)	9.55	16.59	24.63	9.51	16.27	23.93
SuccessProb (direct)	0.91	0.87	0.85	0.91	0.87	0.86
SuccessProb (proposed)	0.87	0.84	0.85	0.86	0.84	0.85
R	$\frac{95}{100}$	$\frac{88}{100}$	$\frac{43}{100}$	$\frac{100}{100}$	$\frac{98}{100}$	$\frac{80}{100}$
CostRed	40.5 %	35.1 %	31.6 %	37.4 %	31.4 %	28.0 %
CompTime	2.4 ms	4.2 ms	12 ms	3.9 ms	9.4 ms	37 ms
simulation area $[m]$	500 × 800					
number of nodes	100			150		
k	2	3	4	2	3	4
Costs (direct)	18.56	29.27	40.32	17.99	27.84	38.42
Costs (proposed)	11.82	20.29	28.57	11.90	19.98	28.40
SuccessProb (direct)	0.89	0.87	0.85	0.89	0.88	0.86
SuccessProb (proposed)	0.85	0.85	0.85	0.86	0.85	0.85
R	$\frac{100}{100}$	$\frac{95}{100}$	$\frac{85}{100}$	$\frac{100}{100}$	$\frac{99}{100}$	$\frac{96}{100}$
CostRed	36.3 %	30.7 %	29.1 %	33.9 %	28.2 %	26.1 %
CompTime	4.4 ms	12.6 ms	39 ms	4.7 ms	16.3 ms	65 ms

6.4 Analysis and simulation results

In this section we analyse the behaviour of our communication algorithm with respect to its complexity and its efficiency.

6.4.1 Complexity

To determine the algorithm's complexity, we refer to the pseudo-code definition found in Section 6.3.2. The key aspect of the first stage is the selection of new nodes, which are split into groups g_1 and g_2. For this purpose, our connectivity value $f_c(n_j)$ is calculated for each TA node n_j, which can be contacted within the time $maxT$. Furthermore, the knowledge of each node is assumed to be restricted to its c-hop neighbourhood. Assuming a network with a maximum node degree d, in which every node is a TA node and the number of $f_c(n_j)$ values that must be calculated to add a node to g_1 or g_2 (line 4 or 6) is $min\left\{d \cdot \frac{(maxT/2)\cdot((maxT/2)+1)}{2}, d \cdot \frac{(c/2)\cdot((c/2)+1)}{2}\right\}$. The operations in line 3, 8 and 9 have a constant complexity. Since these calculations must be executed up to $2k$ times during the "while loop" starting at line 2, the complexity of the first stage can be expressed as $min\left\{O(maxT^2\, k), O(k\, c^2)\right\}$.

The second stage consists of step A (lines 18 to 22) for determining a new route for the remaining TA nodes, and step B (line 23) for reducing the number of TA nodes. In step A, all possible single loops with a maximum length of k are determined. This calculation is performed once for all loops of length i, $i < k$. Each removed node can be deleted from this set to achieve the single loops for $\tilde{k} < k$. The complexity to determine the initial i-loops for $i \leq k$ is $O(d^k)$, as each of the k TA nodes is assumed to have at most d neighbours. Step B requires a single calculation for each of the $\tilde{k}$ remaining nodes, yielding a complexity of $O(\tilde{k})$. The overall complexity of the second stage is therefore $\sum_{\tilde{k}=1}^{k} O(\tilde{k}) = O(\frac{k\cdot(k+1)}{2}) = O(k^2)$.

To combine multiple single loops to one splitting loop, the expectation value for the number of reachable nodes E_N is calculated for all single loops. The number of single loops cannot exceed d^k, and the calculation for the cost expectation value is linear to the number of nodes contained in the loop. In preparation for the final combination to a splitting loop, the single loops are sorted by their E_N values

with an insertion sort with quadratic complexity. Consequently, the complexity for calculating and sorting the E_N values is $O(d^k k^2)$. Finally, a subset of the single loops is combined to the resulting splitting loop. In the worst case, all d^k single loops are chosen successively in the sorted order, yielding constant complexity. The combination of single loops to the resulting splitting loop has a complexity of $O(d^k)$. Thus, the complexity for stage 2 is $O(k^2) + O(d^k) + O(d^k k^2) + O(d^k) = O(d^k k^2)$.

Stage three consists of selecting the best configuration that has occurred during the first and second stages. Since the first stage creates at most $|g_1|$ configurations, and any reduction of the nodes in stage two results in a single new configuration, the number of configurations which are created in the first two stages is linear in k (line 30). Every examination of a configuration requires the reduction of nodes in g_2 (line 31 to 34). As the number of nodes in g_2 is at most k, the complexity for the third stage is $O(k^2)$. Consequently, the overall complexity of the algorithm is $min\left\{O(k \cdot maxT^2), O(k \cdot c^2)\right\} + O(d^k k^2)$. In infrastructure-less tactical networks consisting of usually no more than 150 nodes, $d = 6$ is a reasonable choice. Additionally, in many MANETs the choice of k is typically small [85] and so our algorithm remains feasible within such networks.

6.4.2 Efficiency

To examine the efficiency of our algorithm, we performed 4 test series, each consisting of 100 different network topologies. Our simulations consisted of two groups of 50 and 75 nodes (respectively) in a 400 m $\times$ 500 m area, and of two groups of 100 and 150 nodes (respectively) in an 800 m $\times$ 500 m area. The transmission range of the nodes was set to 100 m, and the topologies were created by randomly generating the nodes' positions within the defined areas. We determine initial TA nodes by a 1-hop cluster algorithm, in which nodes with a high number of neighbours initially

qualify for TA selection [117]. Additionally, we set the reliability of the connections between TA nodes, and the probability that a given TA node participates, to 0.95, i.e $p_j = 0.95$ and $p^{(i)} = (0.95)^i$. We also configured our simulations so that the initiator TA node knows all other TA nodes in its 6-hop neighbourhood ($c = 6$). Furthermore, we set the $minProb$ to 0.8, and the desired maximum execution time for the protocol run to $maxT = 10$ units of time. Furthermore, the algorithm is configured to optimise for minimum communication overhead, i.e., $r_j = f_c$.

Table 6.1 shows the results of our test series. R is the ratio of topologies, which afforded the possibility to reach k nodes. $CostRed = (Costs(direct) - Costs(algo))/Costs(direct)$ shows the expected cost reduction of the proposed routing strategy, compared to the basis case of contacting each TA node independently. Furthermore, $CompTime$ shows the time necessary to complete a protocol. The expected costs (i.e., the number of transmissions) and the probability ($SuccessProb = P^{(L)}(k)$) for successfully reaching k nodes in the time window $maxT$, are compared for the different routing strategies.

For all three values of k, our algorithm achieves a greater cost reduction for a smaller number of nodes. This can be explained by the density of TA nodes in the network; the longer the inter-TA node distances become, the greater the cost reduction achieved by the algorithm's routing strategy.

6.5 Summary

Distributed protocols for MANETs have to cope with Byzantine behaviour and unreliable communication links. Whilst many existing security protocols for TA services build on broadcast techniques and network flooding, we have shown the potential for more reliable and significantly more efficient routing strategies. We

have investigated the use of overlay networks (optimising for certain configurable properties) and partial re-starting, and developed an algorithm for enhancing the efficiency and robustness of these computations. Our simulation results demonstrate significant energy efficiency improvements for small to medium-sized networks (50 to 150 nodes). These efficiency gains may go a long way to ensuring the longevity of TA security services within a network.

Chapter 7

Path authentication

Contents

In this chapter we present a lightweight probabilistic path authentication scheme to detect and diagnose routing misbehavior in MANETs.

7.1 Introduction

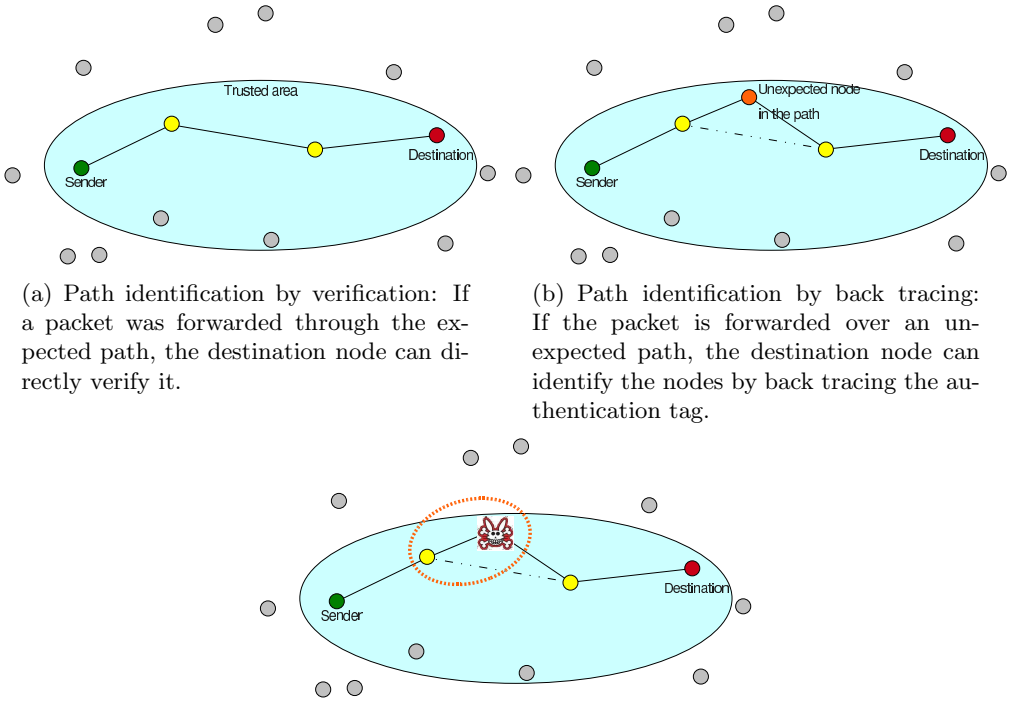
The security of routing protocols against a variety of attacks, including worm hole, impersonation and falsification attacks, has sparked the interest of the research community in recent years. In Section 2.5.4 we gave an overview on secure routing protocols for MANETs, which typically rely on digital signatures, i.e., costly public key cryptography. In this chapter we make a contribution to the security of routing protocols that is based on symmetric key cryptography and is resilient against active Byzantine attackers.

We present a lightweight probabilistic path authentication scheme allowing us to detect and diagnose routing misbehaviour in MANETs. The goal of path authentication is to verify the conformance of the path traversed by a packet with the path prescribed by the underlying routing protocol and detect (and identify) misbehaving nodes in the event of non-conformance. In particular, we focus on incorrect packet forwarding behaviour for the following reasons. First, a malicious node may violate the path prescribed by the routing protocol (e.g., shortest path or trusted path) to interrupt critical data flows or divert traffic to perform timing and traffic analysis attacks. Second, many route falsification attacks (e.g., grey hole or worm hole) result in incorrect packet forwarding behaviour. Third, misconfigured nodes often lead to incorrect packet forwarding behaviour.

We introduce a new cryptographic primitive, *composite Message Authentication*

Code (composite MAC), which forms the basis of our lightweight probabilistic path authentication scheme. In our scheme, composite MACs can have any length starting from one bit. Rather than attempting to authenticate the path traversed by each packet, the proposed scheme amortises the cost of path authentication over a sequence of packets that traverse the same path, while allowing the recipient to collectively authenticate (with high probability) the path traversed by these packets. Besides the authentication of an expected path (Figure 7.1(a)), our scheme facilitates the identification of the path, even when it deviates from the intended one (Figure 7.1(b)). Furthermore, our proposed approach supports the detection of malicious nodes (Figure 7.1(c)) that do not follow the prescribed path authentication scheme correctly, i.e., change the authentication tag in an unintended way. We present a detailed security analysis that shows the detection and diagnostic capabilities of our scheme.

We also present a detailed quantitative analysis that captures various tradeoffs between a resource constrained MANET (e.g., mobility and mean lifetime of a path or size of authentication tags) and the desired security properties (e.g., probability of correct path authentication, probability of diagnosing and pin-pointing misbehaving node(s) in the event of an authentication failure). We show how the number of bits in an authentication tag needs to be chosen to achieve detectability of malicious nodes with a desired probability for a given network size and a given range of route lengths. We argue that the computation costs of identifying the path and the probability of detecting misbehaving nodes compete in an optimisation problem. The results from our quantitative analysis show that a stream of ten packets carrying an eight bit MAC each is sufficient to authenticate a path of length five, and to detect misbehaving nodes (if present) with high probability. These results show that the probabilistic path authentication scheme can operate even on short-lived paths, which are common in MANETs.



(a) Path identification by verification: If a packet was forwarded through the expected path, the destination node can directly verify it.

(b) Path identification by back tracing: If the packet is forwarded over an unexpected path, the destination node can identify the nodes by back tracing the authentication tag.

(c) Detection of a Byzantine node: A malicious node might overwrite the authentication tag of the packets with random bits to avoid back tracing. This node can be identified with an accuracy of up to two nodes (dotted circle).

Figure 7.1: Identification and detection capabilities.

7.2 Background

Recently, several research proposals have used cooperative network monitoring based on root cause analysis techniques to detect malicious and faulty nodes in networks. Cooperative monitoring techniques range from physical layer power estimation for detecting jamming attacks [149][64], and MAC layer misbehaviour detection [108][81] to routing layer faults and anomaly detection [128]. However, to date, all cooperative root cause analysis techniques assume that the monitors are honest, what is not a reasonable assumption in Tactical MANETs.

Boldyreva *et al.* [23] introduced the primitive of an ordered multi-signature scheme, which allows signers to attest to a common message as well as to the order

in which they signed it. The benefit of Boldyreva's scheme compared to previous similar work on multi-signatures is that it does not require synchronised clocks or a trusted first signer. Boldyreva focuses on path authentication in the Internet as the main application of the scheme. Pairing-based signature schemes (as Boldyreva's) have a signature size of typically 60 bytes, which is still small compared to other public key-based signature schemes. Since the typical packet size is 1500 bytes, in wired as well as in wireless communication, the additional communication overhead caused by the 60 byte signature is approximately 5 % (for a 1200 byte payload). We note that most nodes in a MANET are battery powered and thus severely constrained. Hence, while this additional communication overhead might be feasible for the Internet, decreasing the lifetime of a MANET by 5 % appears to be unreasonable. Furthermore, performing elliptic curve operations on each forwarding node for each packet imposes a computational overhead, which is infeasible for devices with limited computational capabilities and battery power.

7.3 Problem definition

In this section we outline the design requirements that our path authentication scheme must satisfy, describe the assumptions that we make about our scheme and outline our adversary model.

7.3.1 Design requirements

- **Unforgeability:** We require the path authentication scheme to be unforgeable by any number of misbehaving nodes on the path.
- **Path identification:** We require a scheme that facilitates to provably identify the nodes on the path even if the path deviates from the expected one, namely,

the path prescribed by the routing protocol. Path identification therefore includes path authentication, since we consider path authentication as the proof of the nodes' identities of an expected path.

- **Detection of misbehaving nodes:** We require a scheme that facilitates detection of misbehaving nodes, including those nodes that attempt to strategically deviate from the path authentication scheme (see Section 7.3.3).
- **Computational efficiency:** We require the scheme to be lightweight and flexible to support a wide range of devices ranging from handheld devices (e.g., PDAs) to laptops.
- **Communication overhead:** We require that the scheme adds at most a few additional bits to each packet. We call this field of additional bits the authentication tag, or *tag* for short.
- **Part of routing:** We require a scheme that blends in the routing protocol: no nodes outside the route shall be involved and no additional packets shall be sent.

7.3.2 Assumptions

- **Symmetric key infrastructure:** We assume that the destination node shares a symmetric key with the source and with each intermediate node on the route to the destination node. Key distribution schemes that require minimal storage and no communication overhead to calculate a shared key, include non-interactive key distribution schemes, as we proposed in Chapter 5. We also assume that the nodes can efficiently perform symmetric key operations as well as compute a collision resistant hash function and a pseudorandom function.

- **Routing protocol:** We assume that the coalition MANET uses a source routing protocol, i.e., the entire path is determined by the source (or the destination) node. Source routing is commonly used in the Internet (e.g., BGP) and MANETs (e.g., DSR, AODV) to support policy-based routing.

7.3.3 Adversary model

We distinguish between two types of attacks that target the path authentication protocol itself:

- **Selfish nodes:** A node is selfish if it forwards packets correctly but ignores the path authentication protocol (e.g., to save energy), i.e., the node does not change the authentication tag as required by the protocol.
- **Byzantine nodes:** A node is Byzantine if it modifies the tag in a way that deviates from the path authentication protocol (e.g., attempts to forge the path, overwrite with random content, etc.).

We want our path authentication scheme to be robust against any number of selfish and Byzantine nodes (Section 7.3.1) in the path. We assume the source and the destination node to be honest.

7.4 Metric-based path authentication algorithm

In this section we introduce our probabilistic path authentication scheme, which uses composite MACs, an extension of aggregate MACs introduced by Katz and Lindell [74] for message authentication. Our scheme allows to verify the conformance of the path traversed by a packet with the path prescribed by the routing protocol; and

to detect (and to identify) misbehaving nodes in the event of non-conformance. We exploit the ability of Katz's scheme to sequentially aggregate several MACs into a constant size authentication tag, while significantly shortening the tag size (say to 1 to 8 bits). We note that short tags result in probabilistic verification; for example, a verified tag of length 4 can only ensure authenticity with a probability of $\frac{15}{16}$. However, the proposed scheme extracts its strength by aggregating the information contained in multiple authentication tags that are embedded in a stream of packets. The proposed scheme is agnostic to packet losses and out-of-order packet arrivals; only the total number of packets used for the authentication is of interest. Hence, composite MACs are especially useful in a MANET setting where communication is unreliable and highly expensive.

7.4.1 Composite MACs

We first recall the aggregate MAC scheme [74]. We use $k_{i,d}$ to denote the shared key between node i (in the path) and node d (the destination node).

Definition 5 (Aggregate Message Authentication Code) *Let* Mac *be a pseudorandom MAC that takes a key $k_{i,d}$ and the actual message m (the rest of the packet that excludes the authentication tag) as input;* tag *is the authentication tag of the same length as* Mac.

- ***Initialisation:*** The sender sets

$$\mathsf{tag} = \mathsf{Mac}_{k_{s,d}}(m) \ ,$$

  where $k_{s,d}$ is the shared key between the sender s and the destination node d. The sender forwards tag and the message m.

- ***Aggregation:*** On input m and tag, a node i sharing the key $k_{i,d}$ with the

destination node, computes

$$\mathsf{tag} = \mathsf{tag} \oplus \mathsf{Mac}_{k_{i,d}}(m) \ .$$

Node i forwards tag and the message m.

- ***Verification:*** On input m, tag and an expected set I of nodes that aggregated their MAC to tag (including the sender), the destination node d verifies:

$$\mathsf{tag} = \bigoplus_{i \in I} \mathsf{Mac}_{k_{i,d}}(m) \ .$$

Definition 6 (Composite Message Authentication Code) *Let* Mac *be a pseudorandom MAC that takes a key* $k_{i,d}$ *and the actual message* m *as input.* tag *is the authentication tag of the same length as* Mac. *Composite MAC extends aggregate MAC by defining the three composition operators Aggregate, Overwrite and KeepIdentical. Nodes in the path pseudo-randomly choose a composition operator that is applied for the authentication tag. We now describe the composite MAC scheme; the role of these composition operators will become evident in the subsequent sections.*

- ***Initialisation:*** The sender sets

$$\mathsf{tag} = \mathsf{Mac}_{k_{s,d}}(m) \ ,$$

where $k_{s,d}$ is the shared key between the sender s and the destination node d. The sender forwards tag and the message m.

- ***Composition:*** On input m and tag, a node i sharing the key $k_{i,d}$ with the destination node, computes

$$\mathsf{tag} = \mathsf{tag} \circ \mathsf{Mac}_{k_{i,d}}(m) \ .$$

Node i forwards tag and the message m. *The composition operator* $\circ$ *can be defined as Aggregate, Overwrite, or KeepIdentical:*

- ***Aggregate:*** $\mathsf{tag} \circ \mathsf{Mac}_{k_{i,d}}(m) = \mathsf{tag} \oplus \mathsf{Mac}_{k_{i,d}}(m)$,
- ***Overwrite:*** $\mathsf{tag} \circ \mathsf{Mac}_{k_{i,d}}(m) = \mathsf{Mac}_{k_{i,d}}(m)$,
- ***KeepIdentical:*** $\mathsf{tag} \circ \mathsf{Mac}_{k_{i,d}}(m) = \mathsf{tag}$,

- ***Verification:*** On input m, tag and an expected ordered set I of nodes that modified tag (including the sender), the destination node d verifies:

$$\mathsf{tag} = \bigcirc_{i \in I} \mathsf{Mac}_{k_{i,d}}(m) \ .$$

7.4.2 Detection of misbehaving nodes

In this section we informally discuss the detection capabilities of the composite MAC scheme. A detailed security analysis is presented in Section 7.5.

Byzantine nodes. While an aggregate MAC, as defined in Definition 5, can be used for path authentication, it does not support detection of Byzantine nodes. For instance, a Byzantine node can easily subvert the aggregate MAC scheme by overwriting the tag with random bits (see Byzantine nodes in Section 7.3.3). Since the remaining nodes on the path would aggregate their MACs with a random tag, the resulting tag would still remain random, and therefore be of no use for the destination node. The composite MAC scheme defeats Byzantine nodes using the composition operator *Overwrite* as follows. Honest nodes positioned between the misbehaving node and the destination node may overwrite the tag with their MACs as part of the composite MAC scheme, thereby allowing us to detect the last Byzantine node in the path with non-trivial probability. The key intuition is that even if a misbehaving node i_j in a path $\{s, i_1, i_2, \ldots, i_r, d\}$ $(j < r)$ replaces the tag with random bits,

benign overwritings by subsequent nodes $\{i_{j+1}, \ldots, i_r\}$ allow the recipient to detect the misbehaving node i_j.

Selfish nodes. A composite MAC, as defined in Definition 6, is agnostic to selfish nodes on the path. Recall that a node is selfish (see Section 7.3.3) if it simply ignores the path authentication scheme, i.e., leaves the tag unchanged. Since selfish nodes put no information at all in the authentication tag, evidence about their existence in the path has to be provided by other nodes. In order to detect selfish nodes, we incorporate the information about the respective prior node $i-1$ as an additional parameter in the MAC. We use F to denote a pseudorandom function that takes the message m, the key $k_{i,d}$ and the identifier ID_{i-1} (of the previous node) as input, and outputs a unique string of the same length as tag:

$$\mathsf{Mac}_{k_{i,d}}(m, ID_{i-1}) = F(m, k_{i,d}, ID_{i-1}) \ . \tag{7.1}$$

Thus, if a node that was expected to be part of a path did not aggregate/overwrite its MAC to an authentication tag when it was expected to, the destination node can identify the selfish node by the MAC of the subsequent node.

7.4.3 Back tracing

In this section we describe our back tracing mechanism, which identifies the nodes on the path traversed by a stream of packets. We recall that back tracing is used when the packet takes an unexpected path, i.e., tag verification failed, indicating that the path traversed by the packets did not conform to the expected path. We first describe a naive and inefficient approach to back tracing. We then show that one can suitably tune the composite MAC scheme to achieve more efficient solutions that can scale with the size of the network.

Let $\mathcal{S}$ denote the set of nodes that are potentially contained in the path. In the absence of any additional information, $\mathcal{S}$ includes the set of all nodes (all participating organisations) in a coalition MANET. Back tracing essentially works by postulating a hypothesis (a plausible path taken by the packet(s)) and corroborating the hypothesis against evidence (a collection of authentication tags on these packet(s)). It is easy to see that the number of such hypothesis (number of plausible paths) is combinatorial in $\mathcal{S}$ in the worst case. To keep the complexity of back tracing low, we use two enhancements.

First, we pseudo-randomly choose only a small subset of nodes on the path to aggregate or overwrite an authentication tag. We ensure that the choice of a node to aggregate, overwrite or keep an authentication tag identical, is known by the respective forwarding node and the destination node, and must not be known by any other node in the network. This approach significantly decreases the number of possible honest nodes that modify the authentication tag, thereby decreasing the cost of identifying an unexpected path by back tracing. At the same time, it is not possible for a bad node to selectively misbehave (and to avoid detection), since it cannot a priori guess the choice of composition (aggregate/overwrite/keep identical) exercised by the good nodes on the path.

We use the parameters p and q to denote the fraction of sub-tags that are modified by aggregation and overwriting, respectively. Consequently, $1 - p - q$ denotes the fraction of sub-tags that are kept identical by a node. To achieve these properties, we let a node i aggregate its MAC to the tag of a packet if:

$$2^{-\lambda} \cdot PRF(packetID, k_{i,d}) \leq p \ ,$$

overwrite the tag with its MAC if:

$$p < 2^{-\lambda} \cdot PRF(packetID, k_{i,d}) \leq p + q \ ,$$

and keep it identical otherwise. PRF is a publicly known pseudorandom function whose output is a non-negative integer of length λ bits, and $k_{i,d}$ is the shared key between node i and the destination node. The packet identifier $packetID$ can be any part of the packet that uniquely defines the packet. Depending on the routing protocol, this could be a sequence number or the timestamp on the packet. Using $packetID$ allows a node i to pseudo-randomly change the choice of composition on a per-packet basis.

As our complexity analysis in Section 7.6.3 shows, a careful choice of the parameters p and q reduces the complexity of back tracing to $poly(|\mathcal{S}|)$[1]. Further, in large MANETs the set of nodes that might participate in a path can be restricted to the nodes that are within a certain distance from the source and destination node.

Second, to enhance the efficacy of back tracing, we argue that using c_n tags of length n/c_n is superior to using one tag of length n. The key intuition here is as follows. In a composite MAC scheme, each verifiable tag serves as evidence for a subsequence of the path traversed by the packet. The table in Figure 7.1 shows the evidence encoded in sample tags attached to packets that traversed the same path. It is easy to see that increasing the *Overwrite* probability q decreases the chances of authenticating long paths. However, in the absence of a non-zero q, the scheme cannot tolerate Byzantine nodes. Given $q > 0$, using multiple sub-tags ($c_n > 1$) allows each sub-tag to serve as an evidence for different subsequences of the path (e.g., $\{i_4, i_5\}$ from tag_2, $\{i_6\}$ from tag_3 and $\{i_3, i_5\}$ from tag_4), thereby enhancing the efficacy of back tracing without compromising detection (of Byzantine nodes). To this end,

[1] $O(|\mathcal{S}|^3)$ under typical parameter settings.

the respective MAC $\mathsf{Mac}_{i,d}$ of length n is divided in c_n MACs $\mathsf{Mac}_{i,d,j}, j = 1, \ldots, c_n$ of length n/c_n such that $\mathsf{Mac}_{i,d} = \mathsf{Mac}_{i,d,1}|\mathsf{Mac}_{i,d,2}|\ldots|\mathsf{Mac}_{i,d,c_n}$ (| is the concatenation operator). A detailed quantitative analysis in Section 7.6 shows the tradeoffs between the number of sub-tags and the efficacy of back tracing and detection.

Table 7.1: Evidence collection.
$\{s, i_1, i_2, \cdots, i_6, d\}$ = path traversed by the packet; i_2 is a Byzantine node; Actions taken by nodes on the path: A = aggregate, O = overwrite, O_r = overwrite with random bits (Byzantine node), K = keep identical; Verifiable = No $\Rightarrow$ tag is useless; Verifiable = Yes $\Rightarrow$ evidence column shows the subset of the path that may be (probabilistically) evidenced by the tag. Combining these evidences, the destination may (probabilistically) conclude that either i_1 or i_2 is a Byzantine node.

Sub-tag	i_1	i_2	i_3	i_4	i_5	i_6	Verifiable	Evidence
tag_1	O	O_r	A	K	K	A	No	-
tag_2	K	O_r	A	O	A	K	Yes	$\{i_4, i_5\}$
tag_3	A	O_r	A	O	A	O	Yes	$\{i_6\}$
tag_4	O	O_r	O	K	A	K	Yes	$\{i_3, i_5\}$

7.5 Security

In this section we examine the security properties of the composite MAC, which include the first two design requirements from Section 7.3.1: unforgeability and detection of misbehaving nodes. While unforgeability is inherited from Katz's aggregate MAC scheme, our security analysis emphasises on the detection capabilities of the composite MAC.

7.5.1 Unforgeability and randomness

Katz and Lindell have proven that aggregate MACs are unforgeable[2] under an adaptive chosen-message attack [62]. The attacker in their security model is allowed to have all but one of the shared keys between the nodes aggregating a message and the destination node. The only requirement is that the individual MACs are unpredictable. This holds for any secure (standard) MAC, by definition [74].

Let us examine the key differences between a composite MAC and an aggregate MAC, namely, the composition operators $OverWrite$ and $KeepIdentical$. A composite MAC that is overwritten one or more times is equivalent to an aggregate MAC whose initial value equals the last overwriting. Nodes that keep the composite MAC identical can be ignored for the security analysis. Since the start value of an aggregate MAC can be any MAC, unforgeability under an adaptive chosen-message attack follows directly from Katz's proof for aggregate MACs. While forging an authentication tag with any non-trivial probability that is larger than 2^{-n} is infeasible (where, n is the size of the tag), using short tags (e.g., $n = 4$ bits) does not preclude the possibility of accidental forgery. However, we show in Section 7.5.2 that the composite MAC scheme can combine evidences from $R \geq 1$ tags to detect misbehaving nodes with a probability one as $R \to \infty$.

Besides the unforgeability, a composite MAC used for path authentication needs to be pseudorandom. If an attacker knew whether the former or the latter nodes on the path were to modify (i.e., aggregate or overwrite) the tag, it could selectively overwrite the tag with the goal of avoiding detection and falsely accusing honest nodes in the path. In our composite MAC scheme, the choice of the composition operator is pseudo-random, making such attacks infeasible.

[2]Infeasible for a poly-time adversary to forge an n-bit authentication tag with probability $2^{-n}+\epsilon$, for some $\epsilon > 0$.

7.5.2 Detection of selfish and Byzantine nodes

Unforgeability and randomness of composite MACs ensure that no node except the destination node can learn any information from a received tag or create a valid tag on behalf of other nodes. Given that the authentication tags are unforgeable (in any meaningful manner), a node may follow one of the following three strategies: (a) Honest: follow the protocol correctly, (b) Selfish: leave the tag unchanged when it was required to modify (aggregate or overwrite) the tag, and (c) Byzantine: overwrite the tag with random bits.

Due to the randomness of the composite MAC, the strategies (a), (b) and (c) cannot be selectively applied on packets. Thus, even if a node were to switch between these strategies, it can at best do so randomly. The analysis of the tags, i.e., path identification and detection of misbehaving nodes, is performed over a collection of tags. Thus, analysing several tags will result in plausible evidences about a node's misbehaviour. If a node switches between strategies (a), (b) and (c), this will be reflected in the evidences about this node. The security analysis therefore tolerates nodes which are switching their strategies; the results will simply apply in the ratio the respective strategies were used.

We define the following sets of nodes:

- I: Ordered set of nodes expected in the path.
- I': Ordered set of nodes contained in the path traversed by the packet(s).
- A: *Good* nodes in the network, following the protocol correctly and putting the correct identity of the former node into the composite MAC.
- B: *Byzantine* nodes in the network, modifying the tag in a way that makes it unreadable for the destination node.

- C: *Selfish* nodes in the network that leave the tag unchanged.
- R: Number of packets used for an analysis.

The detection capabilities of the composite MAC path authentication scheme for Byzantine and selfish nodes are expressed in the following lemma.

Lemma 7.5.1 Given a sufficiently large number of composite MACs, one can with high probability identify: (i) the *good* nodes in the path and the *selfish* nodes, prior to a *good* node in the path, and (ii) the last Byzantine node, or one of the selfish nodes that succeeds the Byzantine node, is detected up to two nodes accuracy. Formally, a series of $R \geq 1$ composite MACs as defined in Section 7.4 contains the following information with non-negligible probability P. Furthermore, P converges to 1 for $R \to \infty$. We distinguish the two cases $B \cap I' = \emptyset$ and $B \cap I' \neq \emptyset$.

1. $B \cap I' = \emptyset$:

   Let $L_{\cap}(X, Y) \to Z$ be the function that takes an ordered set X and a set Y as input, and returns a set Z, which contains for each element $x \in X \cap Y$ the element prior to x in X. Then the information contained in a $R \geq 1$ composite MACs is:

   $$A \cap I' \quad \text{and} \quad L_{\cap}(I', A) \cap C \ ,$$

   namely, the set of *good* nodes in the path and the set of *selfish* nodes prior to a *good* node in the path.

2. $B \cap I' \neq \emptyset$:

   Let $PB(X)$ (pop-back) be the function that returns the last element of an ordered set X, and $R_{\cap}(X, Y) \to Z$ be the function that takes an ordered set X and a set Y as input, and returns a set Z which contains for each element $x \in X \cap Y$ the element after x in X. Then the information contained in the

series of composite MACs is:

$$PB(B) \in B \cup C \quad \text{or} \quad PB(R_{\cap}(I', B)) \in B \quad \text{or} \quad (n \in C) \in B \cup C \ ,$$

i.e., the last *Byzantine*, or one of the *selfish* nodes after him, is detected up to two nodes accuracy.

Lemma 7.5.1 shows that, in the absence of a Byzantine adversary, the good nodes can be exactly identified, and selfish nodes can be detected if they are followed by a good node. In the presence of Byzantine nodes, we can localise the misbehaving node to a set of at most two nodes. Since the destination node cannot decide whether a Byzantine or a selfish node is detected in the detection analysis, it has to fear the worst and accuse this node of being Byzantine. This, however, is an incentive for nodes to follow the protocol, since selfish behaviour might be interpreted as Byzantine behaviour.

Proof

1. $B \cap I' = \emptyset$:

   In the absence of a Byzantine adversary, each good node aggregates or overwrites the tag with non-negligible probability $(p + q)$, and will be authenticated if no later node overwrites $(\geq (1 - q)^{|I'|})$ the tag. Thus, each good node on the route $(A \cap I')$, is authenticated with non-negligible probability $\geq (p+q)\cdot(1-q)^{|I'|} > 0$. Consequently, the set $A \cap I'$ is encoded in a single composite MAC with non-negligible probability, say $P > 0$. Since this statement holds for each composite MAC, in a collection of R composite MACs, the probability that $A \cap I'$ is encoded in one of the composite MACs is $(1-(1-P)^R) \xrightarrow{R \to \infty} 1$.

   Each good node i includes the identity of the prior node ID_{i-1} (the node that forwarded the packet to node i) in the MAC as defined in Equation 7.1.

Thus, if a good node aggregates its MAC (which it does with non-negligible probability $p > 0$), this proves the existence of node $i-1$ in the route. If node $i-1$ is also supposed to aggregate or overwrite its MAC (probability $= p \cdot (p+q) > 0$), but it leaves the tag unchanged, then the selfish behaviour of the node is detected. This argument holds for each selfish node that is followed by a good node on the path; hence, $L_{\cap}(I', A) \cap C$ is encoded in a each composite MAC with non-negligible probability, say P. Consequently, the probability that $L_{\cap}(I', A) \cap C$ is encoded in one of the composite MACs is $(1-(1-P)^R) \xrightarrow{R \to \infty} 1$.

2. $B \cap I' \neq \emptyset$:

   If one or several Byzantine nodes on the path overwrite the composite MAC with random content, then the destination node cannot verify the tag unless the tag has been overwritten by a benign node that succeeded the last Byzantine node on the path. Firstly, with non-negligible probability, none of the nodes after the last Byzantine node may benignly overwrite the composite MAC, such that the tag remains unverifiable; such an unverifiable tag indicates the existence of a Byzantine node in the path. Secondly, with non-negligible probability, a node g after the Byzantine node (if there exists one), overwrites the tag benignly. This shows that there is no Byzantine node that succeed node g on the path. However, the node g may be the Byzantine node itself; a Byzantine node could correctly overwrite the tag that it is supposed to overwrite as part of the scheme, but overwrite the remaining tags (that was supposed to be aggregated or kept identical) with random content (see Table 7.2). Thus, by receiving a correctly overwritten tag, the destination node cannot pin-point the Byzantine node; however, it knows that the overwriting node itself or one of the former nodes is Byzantine.

   With an increasing number of packets R, the probability that the first good node say a_1 after the Byzantine node overwrites the tag converges to one. At

this stage, the destination node has obtained the maximum information that it can get about the last Byzantine node b on the path. If there is no selfish node between b and a_1, then the destination node knows, that either a_1 or the prior node b is the Byzantine node. If there are selfish nodes between b and a_1, then the destination node knows, that either a_1 is a Byzantine node or a_1 is correctly following the protocol, and the prior node is selfish. Therefore, the final conclusion is that either a_1 is Byzantine ($PB(R_{\cap}(I', B)) \in B$), or the prior node of a_1 is either selfish or Byzantine ($PB(B) \in B \cup C \quad \text{or} \quad (n \in C) \in B \cup C$).

□

Table 7.2: Two strategies of a Byzantine node.
The strategies yield different conclusions for the destination node: Strategy I results in evidence revealing the Byzantine node, while Strategy II lets the subsequent node appear to be Byzantine. Action denotes the composition operator that a node is supposed to apply; Strategy I and II show the action taken by a Byzantine node instead of the stipulated action; Actions taken by nodes on the path: A = aggregate, O = overwrite, O_r = overwrite with random (Byzantine node), K = keep identical.

Stipulated Action	A	O	K
Strategy I	O_r	O_r	O_r
Strategy II	O_r	O	O_r

7.6 Configuration and results

In this section we first identify the parameters that need to be configured for our path authentication scheme. We continue to determine the probabilities to (a) identify (verify) a path if the packet is sent over the expected path, (b) identify (back trace) the nodes on an unexpected path, and to (c) detect a Byzantine adversary up to two nodes accuracy. The detection of selfish nodes is incorporated in the path

identification, since the required information to detect selfish nodes is contained in the MACs. Based on these probabilities, we then propose a strategy to configure our probabilistic path authentication scheme. Finally, we present results for the probability to identify a path and to detect Byzantine nodes, depending on both the length n of the tag and the number of packets R used for the analysis. Simulation driven experiments were used to validate our quantitative results and the optimality of our configuration settings.

7.6.1 Parameters

Tables 7.3 and 7.4 show various parameters in our probabilistic path authentication scheme, including those that capture tradeoffs between verification, back tracing and detection of misbehaving nodes. We briefly discuss these parameters to clarify their meaning and influence on the scheme.

Table 7.3: Configuration parameters.

n	The length of the authentication tag in bits.
c_n	The authentication tag is divided in c_n subtags.
p	The ratio of tags (sub-tags) to which each node is supposed to aggregate its MAC.
q	The ratio of tags (sub-tags) that each node is supposed to overwrite with its MAC.
R	The number of packets used for an analysis.
d	Maximum back tracing depth.

Table 7.4: System parameters.

$\mathcal{S}$	Set of nodes in the network that potentially change the tag.
r	The length of the route(s).
s_v	Importance of verification.
s_t	Importance of back tracing and the detection of selfish nodes.
s_B	Importance of the detection of Byzantine nodes.

- n: The length of the authentication tag.

  A longer authentication tag yields more accurate results. However, the tag length should be kept short to minimise the communication overhead.

- c_n: The authentication tag is divided in c_n sub-tags.

  As our analysis shows, dividing the tag in sub-tags enhances the scheme's ability to detect misbehaving nodes.

- p: The ratio of tags (sub-tags) to which every node is supposed to aggregate its MAC.

  This parameter mainly influences the ability to back trace the authentication tags.

- q: The ratio of tags (sub-tags) that every node is supposed to overwrite with its MAC.

  This parameter directly controls the probability to detect Byzantine nodes and influences the ability to verify correct paths.

- R: The number of packets used for an analysis.

  An analysis can be performed over any number of received packets (or tags). Evidently, the accuracy increases with the number of packets used for the analysis. Nevertheless, the optimal choice of p and q depends on the number of packets R.

- d: Maximum tracing depth.

  Back tracing composite MACs from a given authentication tag is performed by hypothesizing a plausible path taken by the packet(s) and verifying the hypothesis against tags (evidences). The number of MACs that are aggregated on such plausible paths is limited to d; this in turn keeps the computational complexity small $O(|\mathcal{S}|^{d+1})$ (see Section 7.6.3). However, computational efficiency comes at the cost of disregarding tags that are aggregated at $d+1$ or more nodes. For the remainder of this section we fix the tracing depth to $d = 2$.

Parameters that influence the configuration of the authentication process:

- $\mathcal{S}$: Set of nodes that potentially change the tag.

  Back tracing needs to be based on a set of nodes $\mathcal{S}$ that potentially aggregate or overwrite the tag. This set might be given by all nodes in the neighbourhood of a path, or even by the entire network. The greater $\mathcal{S}$, the higher is the chance that nodes in $\mathcal{S}$ have the same MAC over a short n-bit tag. Depending on the back tracing strategy, ambiguous MACs might not be considered at all for the back tracing, or only with a smaller weight. In our analysis we only consider unique MACs for back tracing, i.e., if two or more nodes have the same MAC that fits to a back traced tag, we ignore the packet instead of counting it as $1/2$ or $1/n, n > 2$ evidence.

- r: The length of the path(s).

  The longer the path, the more difficult it is to authenticate all nodes on the path; and the harder it is to detect Byzantine nodes. We configure the path authentication scheme to optimally support the longest expected route. Verification and detection of misbehaving nodes become exponentially harder with the path length.

- s_v, s_t, s_B: Importance of verification, authentication and the detection of misbehaving nodes.

  The ability to verify, authenticate (back trace) a path and to detect Byzantine nodes compete in the proposed path authentication scheme. The parameters $s_v, s_t, s_B \in [0,1] \subset \mathbb{R} : s_v + s_t + s_B = 1$ are weights for the importance of these competing parameters. One reasonable choice for these parameters is to choose the importance of the respective event by the ratio of its expected occurrence. For instance, s_v could be defined as the ratio of packets which are sent over the expected path and not modified by a Byzantine node, s_t as the ratio of packets which are sent over an unexpected path and not modified

by a Byzantine node, and s_B as the ratio of packets modified by a Byzantine node. The detection of selfish nodes is included in the verification step and therefore not handled separately. Successful verification automatically proves that no selfish node exists on the path, and back tracing reveals the identities of all detectable selfish nodes.

7.6.2 Probabilities for authentication and detection

As mentioned in the description of the parameter R, the optimal configuration of c_n, p and q depends on the number of packets R used for the analysis. In this section we therefore determine, depending on the number of packets R, the probability to successfully verify or back trace, and to detect Byzantine nodes.

7.6.2.1 Path identification by verification

Analysis of an authentication tag starts with the destination node attempting to verify the tag against the expected path. To this end, the destination node calculates the composite MAC from the verification step in Definition 6 and compares it with the authentication tag in the packet.

As described earlier, let n be the length of the tag divided in c_n sub-tags, p the probability that a node aggregates its MAC to the authentication tag, q the probability that a node overwrites the tag with its MAC, and r the length of the path. The probability that a node at position $s \in \{1, \ldots, r\}$ can be verified by one sub-tag, and that the verification does not happen accidently because of the short tag, is:

$$p_v(s) = (q + p) \cdot (1 - q)^{r-s} \cdot \frac{2^{n/c_n} - 1}{2^{n/c_n}} \ . \tag{7.2}$$

Equation 7.2 is derived as follows. The node in position s overwrites it (q) or aggregates its MAC (p) and none of the remaining nodes overwrite it $(1-q)^{r-s}$. The probability of not authenticating a node by a successful verification after examining 1 and R tags is then $1 - p_v(s)$ and $(1 - p_v(s))^R$, respectively. Consequently, the probability to authenticate a node by a successful verification after R packets ($= R{\cdot}c_n$ sub-tags) is $p_{v,R}(s) = 1 - (1 - p_v(s))^{R \cdot c_n}$. Finally, the probability to authenticate all nodes in the path by successful verification is:

$$p_{v,R,r} = \prod_{s \in \{1,\ldots,r\}} p_{v,R}(s) = \prod_{s \in \{1,\ldots,r\}} 1 - (1 - p_v(s))^{R \cdot c_n} \ . \tag{7.3}$$

7.6.2.2 Path identification by back tracing

If the verification fails, i.e., the composite MAC calculated in the verification step in Definition 6 differs from the tag embedded in the packet, the destination node attempts to back trace with the goal of revealing identities of nodes on the unexpected path. To this end, the destination node hypothesizes a plausible path, and verifies whether the composite MAC that corresponds to the plausible path matches the received tag. Thus it will perform the verification step in Definition 6 for each plausible path I' other than the expected path I, and compares it with the received tag. The theoretical complexity of back tracing is exponential in $|I'|$. To reduce computational complexity, back tracing is limited to a depth d, i.e., back tracing is limited to one overwriting followed by at most d aggregations. However, tags that contain $d + 1$ or more aggregations may be ignored by the destination node; thus, some good evidence may be lost. We note that as long as the probability of $d+1$ or more aggregations is small (p^{-d}), then the probability of disregarding good evidence is small. Further, the destination node may heuristically enumerate plausible paths, starting with a hypothesized path I' that slightly varies from the expected path I, to increase the chances of an early hit.

Let n, c_n, r, p, q, and R be defined as before, and $|\mathcal{S}|$ the total number of nodes that potentially aggregate their MAC to the authentication tag (for example, all nodes in the network). Furthermore, let $\tilde{\mathcal{S}} \subset \mathcal{S}, |\tilde{\mathcal{S}}| > 0$ be a set of nodes that aggregated their MAC to a specific tag. Given $\mathcal{S}$ the probability that $\tilde{\mathcal{S}}$ can be uniquely derived is:

$$p_d(|\mathcal{S}|, |\tilde{\mathcal{S}}|, n, d) = \begin{cases} 0, & n < |\mathcal{S}| \text{ or } d < |\tilde{\mathcal{S}}| \\ \prod_{i=1}^{|\tilde{\mathcal{S}}|} \frac{2^n - 2^{i-1}}{2^n - 1}, & \text{else} \end{cases} \tag{7.4}$$

Setting $n = 3$, the set of valid MACs is $\{001, 010, 011, 100, 101, 110, 111\}$. We do not allow $\{0\}^n$ as a MAC, since it would not be traceable. The first MAC in $\tilde{\mathcal{S}}$ can now be any element from these valid MACs, yielding a probability of 1 if $|\tilde{\mathcal{S}}| = 1$. For a tag that consists of aggregated MACs in $\tilde{\mathcal{S}}$ to be traceable, the second MAC needs to be different from the first one, yielding a probability of $\frac{2^3-2}{2^3-1}$ that the tag is traceable. The next MAC must not be identical to the first, or the second or the combination (XOR) of both. Thus the probability that 3 randomly chosen MACs from $\mathcal{S}$ are traceable after aggregating them is $\frac{2^3-2}{2^3-1} \cdot \frac{2^3-2^2}{2^3-1}$.

The probability that the MAC of a node at position $s \in \{1, \ldots, r\}$ can be revealed, i.e., back traced from the authentication tag, and that the verification does not happen accidently, is:

$$\begin{aligned} p_t(s) \;=\; & (1-q)^{r-s} \cdot \frac{2^{n/c_n} - 1}{2^{n/c_n}} \cdot \Big(q \cdot B\left(d, r-s, p\right) \cdot B\left(n/c_n - i, |\mathcal{S}| - r, p\right) \\ & \cdot p_d\left(j+i, i, n/c_n, d\right) + p \cdot B\left(d-1, r-s-1, p\right) \\ & \cdot B\left(n/c_n - 1, |\mathcal{S}| - r - 1, p\right) \cdot p_d\left(j+i+1, i+1, n/c_n, d\right) \Big) \,, \end{aligned} \tag{7.5}$$

where $B(k, m, p) = \sum_{l=0}^{k} p^m (1-p)^{m-l} \binom{m}{l}$ is the cumulative binomial distribution function. The equation is essentially an extension of Equation 7.2. Firstly, q is the probability that the node at position s overwrites the tag. If this happens, then the nodes on the path between s and the destination node must not overwrite the packet.

Furthermore, only a restricted number of nodes have to aggregate their MAC to the tag to keep the tag traceable. The rest of line one therefore calculates the probability that the tag remains traceable after aggregating MACs from the remaining $r - s$ nodes. $p^i(1-p)^{r-s-i}\binom{r-s}{i}$ is the probability that exactly i of the remaining $r - s$ nodes in the path aggregate their MAC to the tag. $p^j(1-p)^{|\mathcal{S}|-r-j}\binom{|\mathcal{S}|-r}{j}$ is the probability that exactly j of the $|\mathcal{S}|-r$ (all but the r from the path) nodes aggregate their MAC to the tag. The sums stop at d and $\frac{n}{c_n}-i$, since p_d would be 0 for greater values. Recall that p_d represents the probability that the tag is traceable. The second term in the equation can be explained similarly under the supposition that the node at position s aggregates its MAC to the authentication tag. The fraction $\frac{2^{n/c_n}-1}{2^{n/c_n}}$ finally is the probability that the tag is not just a random tag, i.e., the analysis of the tag is not a false positive.

Similar to the verification, the probability to authenticate all nodes in the path of length r by back tracing is:

$$p_{t,R,r} = \prod_{s\in\{1,\ldots,r\}} 1-(1-p_t(s))^{R\cdot c_n} \ . \tag{7.6}$$

7.6.2.3 Detection of Byzantine nodes

Let r be the path length, R the number of the tags used for the analysis and p, q the probabilities for aggregating and overwriting the MAC respectively. The probability of identifying a node at position $s \in \{1,\ldots,r-1\}$ that is randomly overwriting the tag (up to a precision of two nodes) by analysing one authentication tag is[3]:

$$p_B(s) = q(1-q)^{r-s-1}\cdot\frac{2^{n/c_n}-1}{2^{n/c_n}} \ . \tag{7.7}$$

[3]Recall from Lemma 7.5.1 that a Byzantine node can at best be identified up to an precision of two nodes.

Equation 7.7 expresses the probability that the node is at position $s+1$, and that none of the remaining nodes overwrites the tag. If the node at position s is corrupting the authentication tag, then the packets that are overwritten by a node between node s and the destination node are correctly embedded in the authentication tag. Once node s has overwritten a tag with its MAC, the destination node knows that the Byzantine node is either node s or one of the prior nodes. Evidently, the boundary between the Byzantine node and the subsequent good nodes in the path gets more precise with the number of analysed packets. The probability to reveal a Byzantine nodes' identity (up to a precision of two nodes) using R authentication tags, is:

$$p_{B,R,r} = 1 - (1 - p_B(s))^{R \cdot c_n} \ . \tag{7.8}$$

7.6.3 Complexity of back tracing

The number of plausible paths I that need to be distinguished for complete back tracing can be determined as follows: on an average $p|\mathcal{S}|$ nodes in the network aggregate their MAC to the composite MAC, and $q|\mathcal{S}|$ nodes overwrite the composite MAC. Let $P \subset \mathcal{S}$ be the set of nodes that has to aggregate, and $Q \subset \mathcal{S}$ the nodes that overwrite the tag. Now, each combination of subsets of P with at most d nodes can be combined with zero or one node from Q, yielding $(\sum_{i=0}^{d} \binom{|P|}{i})(|Q|+1)$ possible combinations. The average number of combinations for a complete back tracing is therefore $(\sum_{i=0}^{d} \binom{p|\mathcal{S}|}{i})(q|\mathcal{S}|+1)$. Thus, the complexity for backtracing is $O(|\mathcal{S}|^{d+1})$ if $p > 0$.

7.6.4 Configuration

In this section we have so far determined the probability distributions for path authentication by verification, back tracing and detection of Byzantine nodes. The

parameters as listed in Section 7.6.1 that need to be configured for the path authentication scheme are n, c_n, p, q, R, and d. Recall that d is the maximum tracing depth and needs to be configured depending on the computational capabilities of the destination node[4]. As discussed in Section 7.6.1, we set $d = 2$ to allow computationally cheap back tracing. Furthermore, the parameters n and R need to be fixed before we can formulate an optimisation problem that results in an optimal setting for the parameters c_n, p and q. The choice of n and R needs to be balanced with the available network resources (e.g., communication overhead determines n, mobility determines mean path life and thus the number of packets R used for analysis) and the need for accuracy in path identification and detection of Byzantine nodes.

We now assume that the number of nodes potentially modifying the tag (which might in the worst case be the whole network size) and the parameter r_{max} is known. Furthermore, the weights s_v, s_t and s_B and the parameters R and n need to be fixed. In order to maximise both the probability to authenticate the path and to detect Byzantine nodes, we propose to determine p, q and c_n by solving the following optimisation problem:

$$\max \{ s_v \cdot p_{v,R,r_{max}} + s_t \cdot p_{t,R,r_{max}} + s_B \cdot p_{B,R,r_{max}} | p, q \in [0,1] \subset \mathbb{R}, n/c_n \in \mathbb{Z} \} \ . \tag{7.9}$$

To approximately solve this optimisation problem in three variables over non-linear polynomial functions, numeric techniques need to be applied. We give an outline of the technique that we used to approximately determine the optimal values for p, q and c_n. The main observation for our approximation strategy is that p has little or no bearing on the probability to detect Byzantine adversaries $p_{B,R,r}$, and only minor influence on the probability to authenticate a node by verification $p_{v,R,r}$. We therefore determine p to maximise the probability to authenticate by back tracing $p_{t,R,r}$ with a seed value of $q = 0.5$. The calculated p value is then used to determine

[4]If back tracing is *offline* then one can perform deeper tracing.

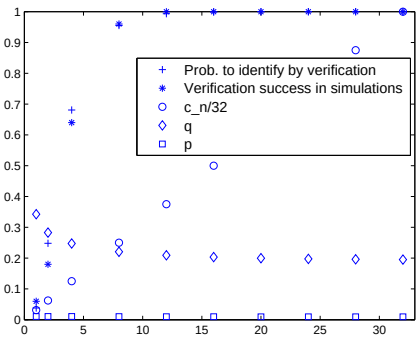
the q that maximises the equation $s_v \cdot p_{v,R,r_{max}} + s_t \cdot p_{t,R,r_{max}} + s_B \cdot p_{B,R,r_{max}}$. We now repeat these two steps iteratively using newer values of q and p respectively at each step. This calculation is performed for each $c_n \mid n \mod c_n = 0$, and the combination of c_n, p, q that maximises Equation 7.9 is chosen as the final result.

Establishing a configuration In order to configure our path authentication scheme for a certain network, firstly either the tag length n or the minimum number of packets R that are expected to be available for the analysis of a path need to be specified. In a MANET with a certain mobility pattern, one could specify R by estimating the lower bound for the number of packets that are typically sent before a path changes due to mobility. Based on different values for n, the probabilities to authenticate the path and to detect Byzantine nodes can then be calculated. n can finally be chosen as the smallest n that satisfies desired authentication and detection probabilities. The resulting values for the sub-tag length c_n and the probabilities p and q for aggregation and overwriting are then used to configure the composite MAC as defined in Section 7.4.1.

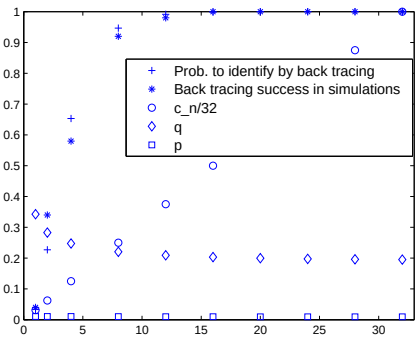
7.6.5 Simulation results

Using Equation 7.9 to calculate near optimal values for p, q and c_n, Figures 7.2 and 7.3 show results for a sample setting. Besides theoretically obtained results, simulation results show the real ratio of nodes that can be identified and detected as Byzantine nodes. For each combination p, q, R and n, i.e., for each marker in the graphs, we ran 100 simulation runs and counted the ratio of tags that allowed us to verify, back trace and detect a Byzantine node, respectively. Figures 7.2 and 7.3 show that the simulation results are close to our theoretical results.

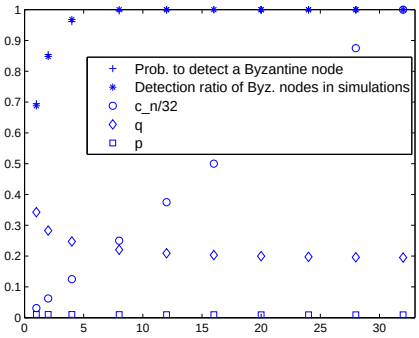
Figure 7.2 shows the calculated values for p, q and c_n and the resulting probabil-



(a) Identification by verification.



(b) Identification by back tracing.

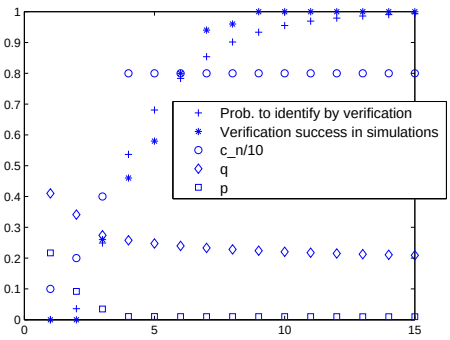


(c) Detecting a Byzantine node.

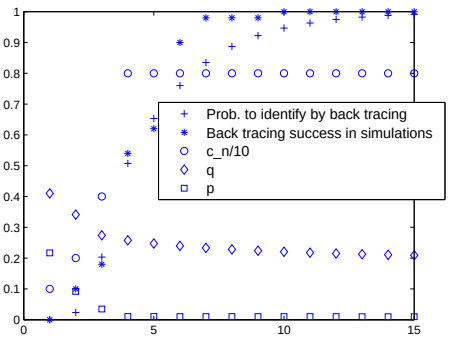
Figure 7.2: Example results for a number of $R = 10$ packets. $\mathcal{S} = 30$, $r_{max} = 5$ and the tag length is varied.

ities for different tag lengths and the route length of $r = r_{max} = 5$. We only show the results for path length $r = 5$, the probabilities for shorter routes would be even higher. The number of nodes potentially modifying the tag is set to $|\mathcal{S}| = 30$, and $R = 10$ packets are used for analysing the tags. The tag length n was set to 1, 2, 4, and multiples of 4 up to 32 and the weights s_v, s_t and s_B are set to $s_v = s_t = s_B = \frac{1}{3}$. Figure 7.3 retains the same setting but varies the number of packets R used for analysis while fixing $n = 8$ bits.

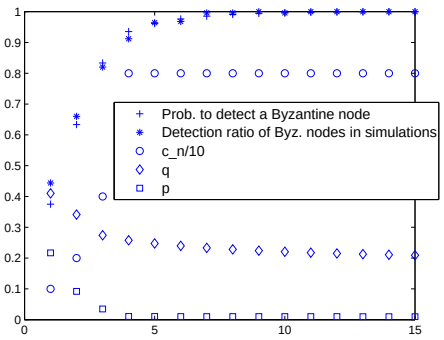
As the Figures 7.2 and 7.3 show, both the length of the authentication tag and the number of packets can be increased to achieve path identification and detection of Byzantine nodes with a desired probability. We also observe that even a very small number of $R = 2$ packets can be sufficient to achieve high probabilities if a



(a) Identification by verification.



(b) Identification by back tracing.



(c) Detecting a Byzantine node.

Figure 7.3: Example results for a tag length of $n = 8$ bits. $\mathcal{S} = 30$, $r_{max} = 5$ and the number of packets R is varied.

sufficiently long tag is used, and vice-versa.

Figures 7.2 and 7.3 show that for sufficiently large R (≥ 4), the optimal value for c_n is n, i.e., the tag is divided into sub-tags each of length one bit; also, the calculated optimal value for p is 0. Note that $p = 0$ means that the composition operator *Aggregate* is seldom used; only $OverWrite$ and $KeepIdentical$ operators are used by the (honest) nodes in the path. Simulations for different settings have affirmed that splitting the authentication into smaller sub-tags increases the probabilities for identification of the path and especially the detection of Byzantine nodes. Informally this result can be explained by the fact that it is better to have many small but probabilistic pieces of evidence than one perfect piece of evidence that rarely occurs.

Finally, we observe that the probability of path identification by verification or back tracing, and the detection of Byzantine nodes are satisfactory (> 0.9) for $R = 10$ or more packets and $n = 8$ bits. This amply demonstrates the efficacy of our path authentication scheme while operating under minimal communication overhead.

7.7 Summary

In this chapter we developed a path authentication scheme feasible for MANETs. The tag length in our probabilistic scheme is scalable, starting with a tag length of 1 bit, and the required computations are cheap (comparable to a hash function). Our scheme uses composite MACs, a new cryptographic primitive which facilitates not only the authentication of paths but also the detection of adversarial nodes. Results show that the combination of evidence from several packets allows to authenticate a path with high probability, even for small tag sizes of only 2–8 bits. The design of our path authentication scheme shows how a probabilistic approach combined with symmetric key cryptography can help to design a scheme that meets the efficiency requirements of MANETs.

Chapter 8

Summary and conclusions

Contents

8.1 Summary

In this thesis we have investigated the development of protocols that satisfy some of the requirements for security, reliability and efficiency presented by MANETs. In particular, we have explored the management of a security architecture within MANETs, which provides the basis for distributed security protocols. Furthermore, we have shown on the example of path authentication how to design protocols that avoid costly computations and how to meet the high security requirements in military MANETs. Whilst the primary driver for this research been military networks,

many of the results obtained are generalisable to other environments that share the constraints presented by these networks.

As preliminary work we created a simulation environment to validate the efficiency and reliability of our security architectures and protocols in military scenarios. This simulation environment (see Chapter 3) includes a radio propagation model and a mobility model, which together facilitate the creation of simulation scenarios in urban environments. Scenarios that incorporate splitting groups and communication interruptions caused by buildings are more challenging than commonly used scenarios, which are characterised by random node mobility in free space. While in random mobility scenarios a partitioning of the network happens only for short time periods, groups in military applications might split for minutes.

In Part I of this thesis we examined the bootstrapping of security architectures within MANETs. Most security protocols require a trusted authority, which does not exist in a MANET per se. Therefore, a set of nodes in the MANET can be assigned as a distributed TA that requires TA nodes to collaborate in order to perform security critical TA computations. The TA nodes either need to be pre-established during the MANET pre-configuration phase or elected/changed dynamically during deployment. To establish a dynamic trust authority, we developed a cluster algorithm in Chapter 4. Efficiency and reliability of this algorithm has been validated in the simulation scenarios defined in Chapter 3. To make our algorithm secure against an active adversary, we incorporated a trust metric into the cluster creation mechanism. To provide stronger security properties for our approach, an adversary model for network protocols would be required. The development of such an adversary model is a challenging but crucial task to provide better security of network protocols, and is discussed in the future work, Section 8.2.4.

A major task of our security architecture for MANETs is the organisation of

cryptographic keys. We have investigated the distribution and efficient storage of symmetric keys in Chapter 5. We proposed two schemes for non-interactive key agreement, which are resilient against a large number of malicious nodes, and due to their computational efficiency suitable for MANETs. For small hierarchies of depth 2 or 3, as can be found in small military MANETs, the size of the keys is only a few KB. This allows online key distribution, as might be required in military networks, when a certain number of nodes got compromised.

In Part II of this thesis we investigated how to efficiently perform distributed computations in MANETs. In Chapter 6 we developed an algorithm for enhancing the efficiency and robustness of distributed trust authority protocols for MANETs. Our algorithm selects a set of TA nodes that are best suited to perform a distributed computation using a suite of metrics for measuring the efficiency and reliability of candidate nodes. Furthermore, our algorithm proposes a routing strategy to contact the selected set of TA nodes. Simulation results showed that the proposed routing strategy considerably reduces the communication cost compared to traditional approaches.

Concluding, in Chapter 7 we developed a path authentication scheme suitable for MANETs. Our scheme is unforgeable and facilitates up to a certain accuracy the detection of malicious nodes on the route. While traditional schemes for path authentication require public key operations, our scheme builds on message authentication codes and therefore only requires symmetric key operations. The use of message authentication codes makes our scheme not only computationally efficient, but also allows the selecion of the length of the authentication tag to any length starting from one bit. Our scheme for path authentication shows how symmetric keys can be effectively used to develop more efficient and flexible algorithms for MANETs.

8.2 Directions for future work

At the end of each chapter we have provided several avenues for future research. However, in addition to these individual pieces of work, a number of key challenges need to be addressed in order to accelerate the widespread adoption of MANETs. We continue with a discussion on improvements for network simulations, including extensions of the mobility model and the ray optical propagation model. MANET simulations will remain an important tool for the evaluation of network protocols, as most of the simulated MANETs are not ubiquitous today, making tests in a real environment infeasible.

Protocols for secret sharing and key distribution, which provide the basis for secure protocols, are yet not able to efficiently cope with dynamic group changes, i.e., merging, splitting, node admission and node departure. Directions for the development of secret sharing and key distribution protocols that facilitate dynamic group changes are given in Section 8.2.3. Protocols that are highly influenced by the network topology such as cluster, routing and revocation algorithms have thus been designed to the best of the designers' knowledge, but lack security proofs in a meaningful adversary model. To this end, we propose the development of an adversary model to provide provable security in these models in Section 8.2.4.

8.2.1 Improving MANET mobility models

In Section 3.1.1 we have introduced CMM, our group mobility model for MANETs. While CMM allows the simulation of complex group movements, as can be seen in our simulation scenarios in Chapter 3, the configuration required for these scenarios is considerable. In this section we discuss an extension of our model that: facilitates the automatic collision avoidance with other nodes and obstacles; allows more

dynamic group changes; and thus allows the creation of complex mobility patterns with less configuration effort than with CMM.

Additional mobility functionalities Williams and Huang [142] recently proposed a mobility model that uses repelling forces to avoid collisions between nodes and with obstacles. We believe that the incorporation of forces is a promising approach to facilitate the creation of very complex mobility scenarios, while keeping the configuration overhead to a minimum. Repelling forces, as already proposed by Williams and Huang, can be used to avoid collisions with other nodes and obstacles. In a group mobility model, forces could additionally be used to organise group formations. We believe that developing a group mobility model solely based on forces facilitates the creation of highly accurate mobility scenarios with minimum configuration overhead.

Imagine for example the formation of a "wedge", as shown in Figure 8.1, that spans preferably an angle of 90° (see also the earlier example in Figure 3.1). A street in a city might be too narrow to keep this formation, i.e., the nodes have to decrease the distances between each other or change their formation to a "wedge" with a smaller angle. If implemented in CMM, this more narrow formation needs to be defined as a separate formation and an explicit formation change needs to be performed. If implemented by a system of forces, the group in formation "wedge" would be automatically squeezed to a more narrow "wedge" as the housewalls get closer. Thus, the formation change would happen automatically and more smoothly.

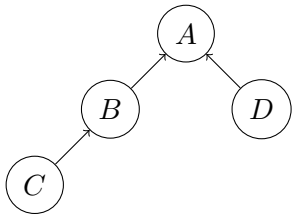


Figure 8.1: A group of nodes in formation "wedge".

To realise formations such as the "wedge" by a force model, nodes can attract or repel each other as they exceed, or fall, below a certain distance. In Figure 8.1 node B could be "attached" by forces to node A as part of the formation, i.e., B should be in an equilibrium if it is located at a certain distance left behind A. Similar techniques are well known in swarm intelligence to model movements of bird flocks. In swarm intelligence, each bird (node) follows simple rules such as holding a certain distance from the birds in front of it and avoiding collisions with nodes to the left and to the right. The bird in front of the flock guides the movement of the whole flock. Following these quite simplistic rules, the bird in front can guide the flock to avoid obstacles and make abrupt direction changes.

Due to the reduction of the configuration complexity and results from swarm-intelligence, we believe that a force model would be well suited to simulate group movements in MANETs.

8.2.2 Increasing our propagation model's accuracy

In Section 3.1.2 we have described the design, implementation and evaluation of a resource-constrained signal propagation model, which demonstrates good fidelity to theoretical and experimental data for our targeted application areas. These include the improvement of routing protocols, which can incorporate situational information such as terrain data and information. The remainder of this section is devoted to discussions of the inclusion of additional factors in our model.

The model described in Section 3.1.2.2 considers transmission via direct line-of-sight, as well as the effects of reflection and deflection. Further factors which can have an appreciable influence on simulation calculations are scattering due to vegetation and absorption by rain. We briefly describe our current analysis and planned implementation steps for these parameters.

Vegetation objects, such as trees or shrubs, form a considerable proportion of topographic objects in many simulation scenarios. In order to describe the scattering of waves caused by vegetation in a ray-optical model, very small polyhedra would need to be used to describe the leaves of trees. However, this naive approach is not feasible, since the polyhedra are assumed to be large compared to the modelled wavelength and because of the computational costs involved. Within the constraints of the ray-optical model, all vegetation objects could also be described by polyhedra with large surfaces. An almost circular shrub, for example, could be described similar to buildings, as a square or hexagon base with an additional parameter for its height. A ray impinging on such a vegetation object would not change its direction, and the power transmitted by the ray would be attenuated, depending on the object size. Following this approach, absorption by vegetation would be taken into consideration, whereas reflection and deflection on vegetation objects would be ignored.

8.2.3 Cryptographic protocols for dynamic MANETs

In Chapter 5 we have introduced our scheme for non-interactive hierarchical key distribution, which facilitates each pair in a group to non-interactively compute a shared secret. A more comprehensive technique for group management is provided by group access control schemes, which we introduced in Section 2.5.2. Schemes for group access control, as proposed by Saxena *et al.* [53, 54], are based on secret sharing, where each node in a group is equipped with one secret share. Based on these secret shares, nodes can not only perform distributed computations, but also use their secret share as a private key in public key protocols, and non-interactively compute pairwise shared secrets. The drawback of secret sharing schemes is their robustness against only a threshold number of compromised nodes in the group. Our non-interactive scheme for key distribution is secure against any number of compromised nodes, if the group is organised in a flat hierarchy.

A challenging task in MANETs is to dynamically organise groups by key distribution or secret sharing schemes, while avoiding a complete re-keying of the group. The support of dynamic group changes is of particular interest in military applications. These changes include node admission, node departure, splitting of groups and merging of groups. First approaches in this line of research include node admission, which can be performed efficiently in secret sharing schemes. Node revocation requires a complete re-keying; the same holds for splitting and merging of groups, which have not gained much attention in the research community so far.

We believe that an extended pre-configuration facilitates the development of more flexible schemes for group organisation. For example, instead of distributing one value, or one polynomial representing the secret share of a node, a set of values or polynomials could be distributed to each node. If, for example, the polynomials span the space of polynomials, then one broadcast could be sufficient to configure all nodes to use a new polynomial as secret share. We believe that such ways of extended pre-configuration can provide more flexible schemes for group organisation.

8.2.4 Adversary model for network protocols

In recent years, game theory has begun to gain attention in the design of cryptographic protocols. [46]. A major benefit of developing protocols in a game theoretic security model is the incorporation of incentive and punishment. Although the nodes in a military MANET usually follow a common mission, they still have to act rationally in their own interest, i.e., selfishly. However, an adversary will delegate his nodes to optimise his overall position, i.e., even sacrifice single nodes if necessary. This makes the design of secure network protocols (such as routing algorithms, cluster algorithms and revocation schemes) for MANETs extremely challenging, since the adversary might use the protocol to his own benefit. We explain the importance

of incentive and punishment in the example of revocation schemes. The problem formulation for revocation schemes is easier than for cluster and routing algorithms, since the decision to make is binary (revoke/do not revoke).

Incentive-based revocation schemes To date, one of the most widely cited methods for achieving revocation in MANETs has been the use of quorum-based decision making, using k-out-of-n threshold signatures [112, 10]. In these schemes, nodes accuse other nodes of malicious behaviour by casting negative votes against a perceived offender. Once a predetermined threshold $k + 1$ of negative votes is achieved, a signature can be reconstructed and the offending node will be considered revoked by other members of the network. Setting this threshold parameter high, whilst intuitively an astute security decision, may inadvertently result in a malicious node never being revoked (as the network density may not support the required level of collaboration). Setting it too low may result in a malicious adversary compromising a relatively small fraction of the total number of nodes and gaining control of the network by being able to revoke at will [34].

To avoid the shortcomings of quorum-based revocation, the concept of node suicide was recently introduced by Clulow *et al.* [38]. Motivated by the observation that many biological systems exhibit behaviour in which individual members of a group are willing to sacrifice themselves to protect the collective (e.g., honeybees sting in response to a perceived threat against the hive), their scheme proposes that a single node can unilaterally revoke another node at the cost of being revoked itself. Unfortunately, for the type of heterogeneous coalition networks envisaged in future military or emergency response scenarios, it may be unreasonable to assume that each node will value the network's utility more than its own. Without sufficient incentive, selfish[1] (rational) nodes will always defer revocation responsibility

[1]Whilst nodes themselves are not capable of higher cognitive processing, we assume nodes are programmed to to maximise their personal utility (or the utility of their group) over a set of constraints.

to others. As was shown in the game-theoretic revocation model in [112], this in turn may result in malicious nodes never being revoked. We believe that designing a revocation scheme in a game-theoretical adversary model enforces the incorporation of incentives to revoke other nodes, yielding a revocation scheme where honest nodes quickly revoke malicious nodes.

Game theoretic security framework The example of revocation has shown the importance of incentive based schemes to minimise the drawback of node selfishness, or put in a positive way, to exploit selfishness. While cryptographic protocols are usually proven to be secure in a specific security model, there exist no such commonly used security models for network protocols such as routing, clustering and node revocation. We see the potential to exploit game theory to build a security framework for network protocols. Protocols that are developed in such a model or framework will automatically incorporate defence mechanisms against attacks and incentivise their use. Investigating the development of a game theoretic security framework or model can therefore help to design more prudent and thus more secure protocols.

8.3 Conclusions

MANETs have the potential to be applicable to a large range of applications that are currently conducted in more traditional networks. In military missions, these applications can provide more comprehensive and reliable information to soldiers, thus helping to minimise risks.

However, communication over a wireless channel opens many possibilities for interception and manipulation. Therefore, the protocols used in military MANETs need to be secure against a wide range of attacks. In this thesis we focused on the

development of secure protocols for MANETs that can be run on power-constrained devices such as handhelds. To provide robustness against compromised nodes, we investigated possibilities to distribute the power for performing security critical computations in a MANET. While protocols for specific applications were introduced in this thesis, the development of secure distributed protocols for a wide range of applications for MANETs is a major task for future research.

Nevertheless, there remain a number of significant obstacles to the widespread use of MANETs. Addressing these challenges is therefore a high priority for future research. Many challenges to the successful large-scale use of MANETs remain.

Bibliography

[1] A. Abdul-Rahman and S. Hailes. A Distributed Trust Model. In *Proceedings of the 1997 Workshop on New Security Paradigms (NSPW '97)*, pages 48–60. ACM Press, September 1997.

[2] C. Adjih, D. Raffo, and P. Mühlethaler. Attacks against OLSR: Distributed Key Management for Security. In *Proceedings of the 2005 OLSR Interop Workshop*, July 2005. `http://perso.crans.org/~raffo/papers/attacks-olsr-dkm.ps.gz`.

[3] A. Amis, R. Prakash, D. Huynh, and T. Vuong. Max-Min D-Cluster Formation in Wireless Ad Hoc Networks. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, pages 32–41. IEEE Communications Society, March 2000.

[4] S. Appel. Lokalisierung von Knoten in Mobilen Ad-hoc-Netzen ohne Zusätzliche Infrastruktur, May 2006. `http://publica.fraunhofer.de/starweb/servlet.starweb?path=pub0.web&search=N-48680`.

[5] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini. Modelling Mobility in Disaster Area Scenarios. In *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07)*, pages 4–12. ACM Press, October 2007.

[6] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini. A Survey on Mobility Models for Performance Analysis in Tactical Mobile Networks. *Journal of Telecommunications and Information Technology (JTIT)*, 2(1):54–61, February 2008.

[7] D. Baker, A. Ephremides, and J. Flynn. The Design and Simulation of a Mobile Radio Network with Distributed Control. *IEEE Journal on Selected Areas in Communications*, 2(1):226–237, January 1984.

[8] C. Balanis. *Advanced Engineering Electromagnetics*. Hamilton Printing Company, 2nd edition, 2002.

[9] S. Balfe, K. Boklan, Z. Klagsbrun, and K. Paterson. Key Refreshing in Identity-based Cryptography and its Applications in MANETS. In *Proceedings of the 2007 IEEE Military Communications Conference 2007 (MILCOM '07)*, pages 1–8. IEEE Communications Society, October 2007.

[10] S. Balfe and S. Reidt. Key Deactivation Strategies in MANETs. In *Proceedings of the 2nd Annual Conference of ITA (AC-ITA '08)*, September 2008. `http://www.usukita.org/files/Page350.pdf`.

[11] J. Bardwell. Converting Signal Strength Percentage to dBm Values, November 2002. `http://www.wildpackets.com/elements/whitepapers/Converting_Signal_Strength.pdf`.

[12] P. Barreto. The Pairing-Based Crypto Lounge. Online; Accessed: May 2009. `http://www.larc.usp.br/~pbarreto/pblounge.html`.

[13] R. Baumann, F. Legendre, and P. Sommer. Generic Mobility Simulation Framework. Online; Accessed: May 2009. `http://gmsf.hypert.net/`.

[14] M. Bechler, H. Hof, D. Kraft, F. Pahlke, and L. Wolf. A Cluster-Based Security Architecture for Ad Hoc Networks. In *Proceedings of the 23rd AnnualJoint*

Conference of the IEEE Computer and Communications Societies (INFOCOM '04), pages 2393–2403. IEEE Computer Sociecty, March 2004.

[15] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS '93)*, pages 62–73. ACM Press, November 1993.

[16] C. Bettstetter. Smooth is Better than Sharp: A Random Mobility Model for Simulation of Wireless Networks. In *Proceedings of the 4th ACM International Workshop on Modeling, Analysis and Smulation of Wireless and Mobile Systems (MSWiM '01)*, pages 19–27. ACM Press, October 2001.

[17] C. Bettstetter, G. Resta, and P. Santi. The Node Distribution of the Random Waypoint Mobility Model for Wireless Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, September 2003.

[18] Z. Biao, X. Kaixin, and M. Gerla. Group and Swarm Mobility Models for Ad Hoc Network Scenarios Using Virtual Tracks. In *Proceedings of the 2004 IEEE Military Communications Conference (MILCOM '04)*, pages 289–294. IEEE Communications Society, November 2004.

[19] K. Bicakci, B. Crispo, and A. Tanenbaum. How to Incorporate Revocation Status Information into the Trust Metrics for Public-Key Certification. In *Proceedings of the 2005 ACM Symposium on Applied Computing (SAC '05)*, pages 1594–1598. ACM Press, March 2005.

[20] K. Blakely and B. Lowekamp. A Structured Group Mobility Model for the Simulation of Mobile Ad Hoc Networks. In *Proceedings of the 2nd International Workshop on Mobility Management & Wireless Access Protocols (MobiWac '04)*, pages 111–118. ACM Press, September 2004.

[21] R. Blom. An Optimal Class of Symmetric Key Generation Systems. In *Proceedings of Advances in Cryptology (EUROCRYPT '84)*, pages 335–338. Springer-Verlag, April 1985.

[22] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-Secure Key Distribution for Dynamic Conferences. *Journal: Lecture Notes in Computer Science*, 146(1):1–23, October 1993.

[23] A. Boldyreva, C. Gentry, A. O'Neill, and D. Yum. Ordered Multisignatures and Identity-based Sequential Aggregate Signatures, with Applications to Secure Routing. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07)*, pages 276–285. ACM Press, October 2007.

[24] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '01)*, pages 213–229. Springer-Verlag, August 2001.

[25] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In *Proceedings of Advances in Cryptology (ASIACRYPT '01)*, pages 514–532. Springer-Verlag, December 2001.

[26] L. Buttyan. Mobility Helps Peer-to-Peer Security. *IEEE Transactions on Mobile Computing*, 5(1):43–51, January 2006.

[27] F. Cai1, H. Fan, L. Rui-xian, H. Liang, and C. Jing. Secure, Redundant, and Fully Distributed Key Management Scheme for Mobile Ad Hoc Networks: An Analysis. *Wuhan University Journal of Natural Sciences*, 11(1):188–192, January 2006.

[28] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *Journal: Wireless Communications & Mobile Computing*

(WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications, 2(5):483–502, September 2002.

[29] CANU Research Group. CanuMobiSim (Mobility Simulation Environment). Institute of Parallel and Distributed Systems (IPVS). University of Stuttgart, 2001. `http://canu.informatik.uni-stuttgart.de/mobisim/`.

[30] G. Caronni. Walking the Web of Trust. In *Proceedings of the 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE '00)*, pages 153–158. IEEE Communications Society, June 2000.

[31] C. Carter, S. Yi, P. Ratanchandani, and R. Kravets. Manycast: Exploring the Space Between Anycast and Multicast in Ad Hoc Networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 273–580. ACM Press, September 2003.

[32] C. Castelluccia, N. Saxena, and J. Yi. Self-configurable Key Pre-distribution in Mobile Ad Hoc Networks. In *Proceedings of the 4th International IFIP-TC6 Networking Conference*, pages 1083–1095. Springer-Verlag, May 2005.

[33] S. Cha and M. Jo. An Energy-Efficient Clustering Algorithm for Large-Scale Wireless Sensor Networks . In *Proceedings of the 2nd International Conference on Advances in Grid and Pervasive Computing (GPC '07)*, pages 436–446. Springer-Verlag, May 2007.

[34] H. Chan, V. Gligor, A. Perrig, and G. Muralidharan. On the Distribution and Revocation of Cryptographic Keys in Sensor Networks. *IEEE Transactions on Dependable and Secure Computing*, 2(3):233–247, September 2005.

[35] C. Chiang, H. Wu, W. LIU, and M. Gerla. Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel. In *Proceedings of the 2007*

IEEE Singapore International Conference on Networks (SICON '07), April 1997. `http://citeseer.ist.psu.edu/chiang97routing.html`.

[36] C.-F. Chiasserini, I. Chlamtac, P. Monti, and A. Nucci. An Energy-Efficient Method for Nodes Assignment in Cluster-Based Ad Hoc Networks. *Jornal on Wireless Networks*, 10(3):223–231, May 2004.

[37] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR), October 2003. `http://www.ietf.org/rfc/rfc3626.txt`.

[38] J. Clulow and T. Moore. Suicide for the Common Good: A New Strategy for Credential Revocation in Self-organizing Systems. *Journal: Operating Systems Reviews — ACM SIGOPS*, 40(3):18–21, July 2006.

[39] S. Corson and J. Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, January 1999. `http://www.ietf.org/rfc/rfc2501.txt`.

[40] G. Crescenzo, M. Fecko, R. Ge, and G. Arce. Securing Weakly-Dominating Virtual Backbones in Mobile Ad Hoc Networks. In *Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '06)*, pages 576–580. IEEE Computer Society, June 2006.

[41] G. Crescenzo, R.G., and G. Arce. Securing Reliable Server Pooling in MANET Against Byzantine Adversaries. *IEEE Journal on Selected Areas in Communications*, 24(2):357–369, February 2006.

[42] G. Danese, Leporati, R. Lombardi, M. Nucita, G. Pedrazzini, and G. Ricotti. An Instrument for the Characterization of Voltage and Temperature Profile in NiCd and NiMH Batteries. In *Proceedings of the 23rd Euromicro Conference: New Frontiers of Information Technology — Short Contributions (EUROMICRO '97)*, pages 178–183. IEEE Computer Society, September 1997.

[43] V. Daza, P. Morillo, and C. Ràfols. On Dynamic Distribution of Private Keys over MANETs. *Journal: Electronic Notes in Theoretical Computer Science (ENTCS '07)*, 171(1):33–41, April 2007.

[44] D. Dhoutaut, A. Régis, and F. Spies. Impact of Radio Propagation Models in Vehicular Ad Hoc Networks Simulations. In *Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks (VANET '06)*, pages 40–49. ACM Press, September 2006.

[45] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.

[46] Y. Dodis and T. Rabin. Cryptography and Game Theory, April 2008. `http://people.csail.mit.edu/dodis/ps/game-survey.ps`.

[47] J. Douceur. The Sybil Attack. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS 02)*, pages 251–260. Springer-Verlag, March 2002.

[48] W. Du, J. Deng, Y. Han, and P. Varshney. A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pages 42–51. ACM Press, October 2003.

[49] A. Enge. Practical Non-Interactive Key Distribution Based on Pairings. In *Proceedings of the 2002 International Workshop on Coding and Cryptography (WCC '02)*, September 2002. `http://eprint.iacr.org/2002/136`.

[50] L. Eschenauer and V. Gligor. A Key-Management Scheme for Distributed Sensor Networks. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pages 41–47. ACM Press, November 2002.

[51] K. Fall and K. Varadhan. The NS Manual. Online; Accessed: May 2009. `http://www.isi.edu/nsnam/ns/ns-documentation.html`.

[52] F. Foroozan and K. Tepe. A High Performance Cluster-Based Broadcasting Algorithm for Wireless Ad Hoc Networks Based on a Novel Gateway Selection Approach. In *Proceedings of the 2nd ACM international workshop on Performanceevaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN '05)*, pages 65–70. ACM Press, October 2005.

[53] N. S. G., Tsudik, and J. Yi. Threshold Cryptography in P2P and MANETs: The Case of Access Control. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 51(12):3632–3649, August 2007.

[54] N. S. G., Tsudik, and J. Yi. Efficient Node Admission and Certificateless Secure Communication in Short-Lived MANETs. *IEEE Transactions on Parallel and Distributed Systems*, 20(2):158–170, February 2009.

[55] R. Ge, G. Crescenzo, M. Fecko, and S. Samtani. Efficient and Secure Indirectaaddress Service Discovery in MANET. In *Proceedings of the 2005 IEEE Military Communications Conference (MILCOM '05)*, pages 1514–1520. IEEE Communications Society, October 2005.

[56] N. Geng and W. Wiesbeck. *Planungsmethoden für die Mobilkommunikation*. Springer-Verlag, 1st edition, September 1998.

[57] R. Gennaro, S. Halevi, H. Krawczyk, T. Rabin, S. Reidt, and S. Wolthusen. Strongly-Resilient and Non-Interactive Hierarchical Key-Agreement in MANETs. In *Proceedings of the 13th European Symposium on Research in Computer Security (ESORICS '08)*, pages 49–65, October 2008.

[58] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. *Journal of Cryptology*, 20(1):51–83, January 2007.

[59] M. Gerharz, C. de Waal, P. Martini, and P. James. A Cooperative Nearest Neighbours Topology Control Algorithm for Wireless Ad Hoc Networks. In *Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN '03)*, pages 1412–417. IEEE Computer Society, October 2003.

[60] M. Gerharz, C. de Waal, P. Martini, and P. James. MStrategies for Finding Stable Paths in Mobile Wireless Ad Hoc Networks. In *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN '03)*, pages 130–139. IEEE Computer Society, October 2003.

[61] globalsecurity.org. Military Operations — Movements. Online; Accessed: May 2009. `http://www.globalsecurity.org/military/library/policy/army/fm/7-8/`.

[62] S. Goldwasser, S. Micali, and R. Rivest. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. *SIAM Journal on Computing (SICOMP)*, 17(1):281–308, April 1988.

[63] G. Hanaoka, T. Nishioka, Y. Zheng, and H. Imai. A Hierarchical Non-interactive Key-Sharing Scheme with Low Memory Size and High Resistance against Collusion Attacks. *The Computer Journal*, 45(3):293–303, 2002.

[64] I. Ho, B. Ko, M. Zafer, C. Bisdikian, and K. Leung. Cooperative Transmit-Power Estimation in MANETs. In *Processings of the 2008 Wireless Communications and Networking Conference (WCNC '08)*, pages 2241–2246. ACM Press, March 2008.

[65] X. Hong, M. Gerla, G. Pei, and C. Chiang. A Group Mobility Model for Ad Hoc Wireless Networks. In *Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '99)*, pages 53–60. ACM Press, August 1999.

[66] R. Hoppe, P. Wertz, F. Landstorfer, and G. Wölfle. Advanced Ray Optical Wave Propagation Modelling for Urban and Indoor Scenarios Including Wideband Properties. *European Transactions on Telecommunications*, 14(1):61–69, January 2003.

[67] R. Hoppe, G. Wölfle, and F. Landstorfer. Fast 3-D Ray Tracing for the Planning of Microcells by Intelligent Preprocessing of the Database. In *Proceedings of the 3rd European Personal and Mobile Communications Conference (EPMCC '99)*, pages 149–154. IEEE Communications Society, March 1999.

[68] J. Horwitz and B. Lynn. Toward Hierarchical Identity-Based Encryption. In *Proceedings of Advances in Cryptology (EUROCRYPT '02)*, pages 466–481. Springer-Verlag, April/May 2002.

[69] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Towards Realistic Mobility Models for Mobile Ad Hoc Networks. In *Proceedings of the 9th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '03)*, pages 217–229. ACM Press, August 2003.

[70] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Real-world Environment Models For Mobile Network Evaluation. *IEEE Journal on Selected Areas in Communications*, 23(3):622–632, March 2005.

[71] M. Jiang, J. Li, and Y. Tay. Cluster Based Routing Protocol (CBRP). Functional Specification. Internet Engineering Task Force Draft, August 1999. `http://tools.ietf.org/html/draft-ietf-manet-cbrp-spec-01`.

[72] D. Johnson, D. Maltz, and J. Broch. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, February 2007. `http://www.ietf.org/rfc/rfc4728.txt`.

[73] D. Joshi, K. Namuduri, and R. Pendse. Secure, Redundant, and Fully Distributed Key Management Scheme for Mobile Ad Hoc Networks: An Analysis. *Journal: Wireless Communications and Networking*, 5(4):579–589, September 2005.

[74] J. Katz and A. Lindell. Aggregate Message Authentication Codes. In *Proceedings of Topics in Cryptology (CT-RSA '08)*, pages 155–169. Springer-Verlag, April 2008.

[75] A. Khalili, J. Katz, and W. Arbaugh. Toward Secure Key Distribution in Truly Ad-Hoc Networks. In *Proceedings of the Symposium on Applications and the Internet Workshops 2003 (SAINT'03 Workshops)*, pages 342–346. IEEE Computer Society, January 2003.

[76] Y. Kim, D. Mazzocchi, and G. Tsudik. Admission Control in Peer Groups. In *Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications (NCA '03)*, pages 131–140. IEEE Computer Society, July 2003.

[77] R. Kohlas and U. Maurer. Confidence Valuation in a Public-key Infrastructure Based on Uncertain Evidence. In *Proceedings of the 3rd International Workshop on Practice and Theory in Public Key Cryptography (PKC '00)*, pages 93–112. Springer-Verlag, January 2000.

[78] J. Kong, Z. Petros, L. Haiyun, L. Songwu, and Z. Lixia. Providing Robust and Ubiquitous Security Support for Mobile Ad Hoc Networks. In *Proceedings of the 9th International Conference on Network Protocols (ICNP '01)*, pages 251–260. IEEE Computer Society, November 2001.

[79] S. Kurkowski, T. Camp, and M. Colagrosso. MANET Simulation Studies: The Current State and New Simulation Tools. Technical report, Colorado School of Mines, February 2005. `http://toilers.mines.edu/pub/Public/PublicationList/CSM-MCS-05-02.pdf`.

[80] T. Kwon and M. Gerla. Efficient flooding with Passive Clustering (PC) in Ad Hoc Networks. *Journal: Computer Communication Review — ACM SIGCOMM*, 32(1):44–56, January 2002.

[81] Y. Law, L. van Hoesel, J. Doumen, P. Hartel, and P. Havinga. Energy-Efficient Link-Layer Jamming Attacks Against Wireless Sensor Network MAC Protocols. In *Proceedings of the 3rd ACM workshop on Security of Ad Hoc and Sensor Networks (SASN '05)*, pages 76–88. ACM Press, November 2005.

[82] T. Lin-Jiun, L. Jen-Chiun, and L. Feipei. SWARM: Secure Wireless Adhoc network Reliance Management. In *Proceedings of the 1st International Symposium on Wireless Pervasive Computing (ISWPC '06)*, pages 1–6. IEEE Computer Sociecty, January 2006.

[83] D. Liu and P. Ning. Establishing Pairwise Keys in Distributed Sensor Networks. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pages 52–61. ACM Press, October 2003.

[84] X. Lu, Y. Chen, I. Leung, Z. Xiong, and P. Liò. A Novel Mobility Model from a Heterogeneous Military MANET. In *Proceedings of the 7th International Conference on Ad-Hoc Networks & Wireless (ADHOC-NOW '08)*, pages 463–474. Springer-Verlag, September 2008.

[85] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks. *IEEE/ACM Journal on Transactions on Networking*, 12(6):1049–1063, December 2004.

[86] H. Luo and S. Lu. Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks. Technical Report TR-200030, Dept. of Computer Science, UCLA, October 2000. `http://reference.kfupm.edu.sa/content/u/b/ubiquitous_and_robust_authentication_ser_730789.pdf`.

[87] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-Securing Ad Hoc Wireless Networks. In *Proceedings of the 7th International Symposium on Computers and Communications (ISCC' 02)*, pages 567–574. IEEE Computer Society, July 2002.

[88] J. Marchesini and S. Smith. Modeling Public Key Infrastructures in the Real World. In *Proceedings of the 2nd European PKI Workshop (EuroPKI '05)*, pages 118–134. Springer-Verlag, June/July 2005.

[89] B. Matt. Toward Hierarchical Identity-based Cryptography for Tactical Networks. In *Proceedings of the 2004 Military Communications Conference (MILCOM '04)*, pages 727–735. IEEE Communications Society, November 2004.

[90] J. Maurer. *Strahlenoptisches Kanalmodell für die Fahrzeug-Fahrzeug-Funkkommunikation.* PhD thesis, Institut fuer Höchstfrequenztechnik und Elektronik, Universität Karlsruhe, July 2005. `http://digbib.ubka.uni-karlsruhe.de/volltexte/1000003404`.

[91] U. Maurer. Modelling a Public-Key Infrastructure. In *Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS '96)*, pages 325–350. Springer-Verlag, September 1996.

[92] T. Moore. Efficient Security Primitives Derived from a Secure Aggregation Algorithm. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, (PerCom Workshops '06)*, pages 251–255. IEEE Computer Society, March 2006.

[93] N. Mushell and T. Camp. iNSpect — a Visualization Tool for MANET Simulations. Online; Accessed: May 2009. `http://www.igd.fhg.de/igd-a8/de/projects/mobsec/inspect`.

[94] M. Narasimha, G. Tsudik, and J. Yi. On the Utility of Distributed Cryptography in P2P and MANETs: The Case of Membership Control. In *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP '03)*, pages 336–346. IEEE Computer Society, December 2003.

[95] W. Navidi and T. Camp. Stationary Distributions for the Random Waypoint Mobility Model. *IEEE Transactions on Mobile Computing*, 3(1):99–108, June 2004.

[96] E. Ochirsuren, L. Indrusiak, and M. Glesner. An Actor-oriented Group Mobility Model for Wireless Ad Hoc Sensor Networks. In *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops (ICDCS '08)*, pages 1017–1021. IEEE Computer Society, June 2008.

[97] D. Page, N. Smart, and F. Vercauteren. A Comparison of MNT Curves and Supersingular Curves. *Journal: Applicable Algebra in Engineering, Communication and Computing*, 17(5):379–392, October 2006.

[98] P. Papadimitratos and Z. Haas. Secure Link State Routing for Mobile Ad Hoc Networks. In *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT '03 Workshops)*, pages 379–383. IEEE Computer Society, January 2003.

[99] V. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '07)*, volume 3, pages 1405–1413. IEEE Press, April 1997.

[100] B. Parno, A. Perrig, and V. Gligor. Distributed Detection of Node Replication Attacks in Sensor Networks. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 49–63. IEEE Computer Society, May 2005.

[101] T. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *Proceedings of Advances in Cryptology (EUROCRYPT '91)*, pages 522–526. Springer-Verlag, April 1991.

[102] C. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *Journal: Computer Communication Review — ACM SIGCOMM*, 24(4):234–244, October 1994.

[103] C. Perkins and E. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '09)*, pages 90–100. IEEE Press, February 1999.

[104] M. Piorkowski, M. Raya, A. Lugo, and J.-P. Hubaux. TRANS: Realistic Simulator for Vanets. Online; Accessed: May 2009. `http://trans.epfl.ch/`.

[105] S. Pleisch, M. Balakrishnan, K. Birman, and R. van Renesse. MISTRAL: Efficient Flooding in Mobile Ad-hoc Networks. In *Proceedings of the Seventh ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '06)*, pages 1–12. ACM Press, May 2006.

[106] N. Potnis and A. Mahajan. Mobility Models for Vehicular Ad Hoc Network Simulations. In *Proceedings of the 44th Annual Southeast Regional Conference (ASRC '06)*, pages 746–747. ACM Press, March 2006.

[107] W. Qi, X. Zhang, and H. Yu. An Improved CEDAR Routing Protocol. In *Proceedings of the 4th International Conference on Computer and Information Technology (CIT '04)*, pages 621–626. IEEE Communications Society, September 2004.

[108] S. Radosavac, J. Baras, and I. Koutsopoulos. A Framework for MAC Protocol Misbehavior Detection in Wireless Networks. In *Proceedings of the 4th ACM workshop on Wireless security (WiSe '05)*, pages 33–42. ACM Press, September 2005.

[109] M. Ramkumar, N. Memon, and R. Simha. A Hierarchical Key Pre-distribution Scheme. In *Proceedings of the IEEE International Conference on Electro Information Technology 2005 (EIT '05)*, pages 6–11. IEEE Computer Society, May 2005.

[110] T. Rautiainen, R. Hoppe, and G. Wölfle. Measurements and 3D Ray Tracing Propagation Predictions of Channel Characteristics in Indoor Environments. In *Proceedings of the 18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '09)*, pages 1–5. IEEE Communications Society, September 2009.

[111] T. Rautiainen, G. Wölfle, and R. Hoppe. Verifying Path Loss and Delay Spread Predictions of a 3D Ray Tracing Propagation Model in Urban Environments. In *Proceedings of the 56th IEEE Vehicular Technology Conference (VTC '02)*, pages 2470–2474. IEEE Communications Society, May 2002.

[112] M. Raya, M. H. Manshaei, M. Félegyhazi, and J.-P. Hubaux. Revocation Games In Ephemeral Networks. In *Proceedings of the 15th ACM conference on Computer and Communications Security (CCS '08)*, pages 199–210. ACM Press, October 2008.

[113] S. Reidt. Topographisches Routing in mobilen Ad-Hoc-Netzen. Master's thesis, Technical University of Darmstadt, September 2006. `http://www.sreidt.com/wp-content/uploads/2009/01/diploma-thesis.pdf`.

[114] S. Reidt, P. Ebinger, and S. Wolthusen. Resource-Constrained Signal Propagation Modeling for Tactical Networks. under submission, 2006.

[115] S. Reidt and M. Srivatsa. Inter-Domain Path Authentication in Tactical MANETs. In *Proceedings of the 26th Army Science Conference (ASC '08)*, December 2008. `http://www.asc2008.com/manuscripts/B/BP-03.pdf`.

[116] S. Reidt and S. Wolthusen. An Evaluation of Cluster Head TA Distribution Mechanisms in Tactical MANET Environments. In *Proceedings of the 1st Annual Conference of the ITA (AC-ITA '07)*, September 2007. `http://www.usukita.org/papers/3020/1569048335.pdf`.

[117] S. Reidt and S. Wolthusen. Efficient Distribution of Trust Authority Functions in Tactical Networks. In *Proceedings from the 8th Annual IEEE SMC Information Assurance Workshop (IAW '07)*, pages 84–91. IEEE Press, June 2007.

[118] S. Reidt and S. Wolthusen. Efficient Trust Authority Distribution in Tactical MANET Environments. In *Proceedings of the 2007 IEEE Military Communications Conference (MILCOM '07)*, pages 1–7. IEEE Communications Society, October 2007.

[119] S. Reidt, S. Wolthusen, and S. Balfe. Robust and Efficient Communication Overlays for Trust Authority Computations. In *Proceedings of the 2009 IEEE Sarnoff Symposium*, pages 1–5. IEEE Communications Society, March 2009.

[120] M. Reiter and S. Stubblebine. Authentication Metric Analysis and Design. *ACM Transactions on Information and System Security (TISSEC '99)*, 2(2):138–158, 1999.

[121] E. Rescorla. Diffie-Hellman Key Agreement Method, June 1999. `http://www.ietf.org/rfc/rfc2631.txt`.

[122] E. Royer, P. Melliar-Smith, and L. Moser. An Analysis of the Optimum Node Density for Ad hoc Mobile Networks. In *Proceeding of the 2001 IEEE In-*

ternational Conference on Communications (ICC '01), pages 857–861. IEEE Communications Society, June 2001.

[123] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairing. In *Proceedings of the 2000 Symposium on Cryptography and Information Security (SCIS '00)*, pages 26–28, January 2000.

[124] N. Saxena, G. Tsudik, and J. Yi. Admission Control in Peer-to-Peer: Design and Performance Evaluation. In *Proceedings of the 1st ACM workshop on Security of Ad hoc and Sensor Networks (SASN '03)*, pages 104–113. ACM Press, April 2003.

[125] N. Saxena, G. Tsudik, and J. H. Yi. Efficient Node Admission for Short-lived Mobile Ad Hoc Networks. In *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP'05)*, pages 269–278. IEEE Computer Society, November 2005.

[126] A. Shamir. How to Share a Secret. *ACM Communications Journal*, 22(11):612–613, November 1979.

[127] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. In *Proceedings of the 4th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '84)*, pages 47–53. Springer-Verlag, August 1985.

[128] M. Srivatsa, B.-J. Ko, A. Beygelzimer, and V. Madduri. Scalable Topology Discovery and Link State Detection Using Routing Events. In *Proceedings of the 2008 Symposium on Reliable Distributed Systems (SRDS '08)*, pages 165–174. IEEE Computer Society, December 2008.

[129] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. *Security for Ubiquitous Computer*, 35(4):22–26, April 2002.

[130] I. Stepanov, D. Herrscher, and K. Rothermel. On the Impact of Radio Propagation Models on MANET Simulation Results. In *Proceedings of the 7th IFIP International Conference on Mobile and Wireless Communications Networks (MWCN '05)*, September 2005. `ftp://ftp.informatik.uni-stuttgart.de/pub/library/ncstrl.ustuttgart_fi/INPROC-2005-37/INPROC-2005-37.pdf`.

[131] I. Stepanov, P. Marrón, and K. Rothermel. Mobility Modeling of Outdoor Scenarios for MANETs. In *Proceedings of 38th Annual Simulation Symposium (ANSS '05)*, pages 312–322. IEEE Computer Society, April 2005.

[132] Y. Sun, W. Yu, Z. Han, and K. Liu. Information Theoretic Framework of Trust Modeling and Evaluation for Ad hoc Networks. *IEEE Journal on Selected Areas in Communications*, 24(2):305–317, February 2006.

[133] B. B. T. T. T. Hamma, T. Katoh. An Efficient ZHLS Routing Protocol for Mobile Ad hoc Networks. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA '06)*, pages 66–70. IEEE Communications Society, September 2006.

[134] G. Theodorakopoulos and J. Baras. On Trust Models and Trust Evaluation Metrics for Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, February 2006.

[135] J. Tian, J. Haehner, C. Becker, I. Stepanov, and K. Rothermel. Graph-based Mobility Model for Mobile Ad Hoc Network Simulation. In *Proceedings of 35th Annual Simulation Symposium (ASS '02)*, pages 337–344. IEEE Communications Society, April 2002.

[136] C.-K. Toh. Associativity-Based Routing for Ad Hoc Mobile Networks. *Journal: Wireless Personal Communications*, 4(2):103–139, 1997.

[137] C. Tuduce and T. Gross. A Mobility Model Based on WLAN Traces and its Validation. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies. (INFOCOM '05)*, pages 664–674. IEEE Computer Society, March 2005.

[138] University of Bonn. BonnMotion: A Mobility Scenario Generation and Analysis Tool, 2002. `http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion`.

[139] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Mobility Helps Security in Ad Hoc Networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MobiHoc '03)*, pages 46–56. ACM Press, June 2003.

[140] K. Wang and L. Baochun. Group Mobility and Partition Prediction in Wireless Ad-Hoc Networks. In *Proceedings of the 2002 IEEE International Conference on Communications (ICC '02)*, pages 1017–1021. IEEE Communications Society, May 2002.

[141] Y. Wang, W. Wang, and X. Li. Distributed Low-Cost Backbone Formation for Wireless Ad Hoc Networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '05)*, pages 2–13. ACM Press, May 2005.

[142] S. Williams and D. Huang. A Group Force Mobility Model. In *Proceedings of the 9th Communications and Networking Simulation Symposium (CNS '06)*, April 2006. `http://www.public.asu.edu/~sawilli3/index_files/GroupForceSimulationFinal.pdf`.

[143] B. Wu, J. Wu, E. Fernandez, and S. Magliveras. Secure and Efficient Key Management in Mobile Ad Hoc Networks. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05)*, pages 288–296. IEEE Computer Society, April 2005.

[144] L. Xiong and L. Liu. Building Trust in Decentralized Peer-to-Peer Electronic Communities. In *Proceedings of the 5th International Conference on Electronic Commerce Research (ICECR '02)*, October 2002. `http://www.mathcs.emory.edu/~lxiong/research/pub/xiong02building.pdf`.

[145] G. Xu and L. Iftode. Locality Driven Key Management Architecture for Mobile Ad-hoc Networks. In *Proceedings of the 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '04)*, pages 436–446. IEEE Computer Society, October 2004.

[146] S. Yi and R. Kravets. Practical PKI for Ad Hoc Wireless Networks. Technical Report UIUCDCS-R-2002-2273, UILU-ENG-2002-1717, Department of Computer Science, University of Illinois at Urbana-Champaign, August 2001. `http://historical.ncstrl.org/tr/ps/uiuc_cs/UIUCDCS-R-2002-2273.ps`.

[147] S. Yi and R. Kravets. Key Management for Heterogeneous Ad Hoc Wireless Networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP '02)*, pages 202–205. IEEE Computer Society, November 2002.

[148] S. Yi and R. Kravets. MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks. In *Proceedings of the 2nd Annual PKI Research Workshop Program (PKI '03)*, pages 65–79. University of Illinois, April 2003.

[149] M. Zafer, B. Ko, and I. Ho. Cooperative Transmit-Power Estimation under Wireless Fading. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '08)*, pages 381–390. ACM Press, May 2008.

[150] M. Zapata and N. Asokan. Securing Ad hoc Routing Protocols. In *Proceedings of the 1st ACM workshop on Wireless security (WiSE '02)*, pages 1–10. ACM Press, September 2002.

[151] M. G. Zapata. Secure Ad Hoc On-Demand Distance Vector Routing. *Journal: Mobile Computing and Communications Review — ACM SIGMOBILE*, 6(3):106–107, July 2002.

[152] W. Zhang, M. Tran, S. Zhu, and G. Cao. A Random Perturbation-Based Scheme for Pairwise Key Establishment in Sensor Networks. In *Proceedings of the 8th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '07)*, pages 90–99. ACM Press, September 2007.

[153] L. Zhou and Z. Haas. Securing Ad Hoc Networks. *IEEE Network Journal, special issue on network security*, 13(6):24–30, November 1999.

[154] P. Zimmermann. *The Official PGP User's Guide*. MIT Press, June 1995.

[155] L. Zongpeng and L. Baochun. Probabilistic Power Management for Wireless Ad Hoc Networks. *Journal: Mobile Networks and Applications*, 10(5):771–782, October 2005.