Authentication and Key Exchange in Mobile Ad Hoc Networks

by

Katrin Hoeper

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Over the past decade or so, there has been rapid growth in wireless and mobile applications technologies. More recently, an increasing emphasis has been on the potential of infrastructureless wireless mobile networks that are easy, fast and inexpensive to set up, with the view that such technologies will enable numerous new applications in a wide range of areas. Such networks are commonly referred to as mobile ad hoc networks (MANETs). Exchanging sensitive information over unprotected wireless links with unidentified and untrusted endpoints demand the deployment of security in MANETs. However, lack of infrastructure, mobility and resource constraints of devices, wireless communication links and other unique features of MANETs induce new challenges that make implementing security a very difficult task and require the design of specialized solutions.

This thesis is concerned with the design and analysis of security solutions for MANETs. We identify the initial exchange of authentication and key credentials, referred to as pre-authentication, as well as authentication and key exchange as primary security goals. In particular, the problem of pre-authentication has been widely neglected in existing security solutions, even though it is a necessary prerequisite for other security goals. We are the first to classify and analyze different methods of achieving pairwise pre-authentication in MANETs. Out of this investigation, we identify identity-based cryptographic (IBC) schemes as well-suited to secure MANET applications that have no sufficient security solutions at this time.

We use pairing-based IBC schemes to design an authentication and key exchange framework that meets the special requirements of MANETs. Our solutions are comprised of algorithms that allow for efficient and secure system set up, pre-authentication, mutual authentication, key establishment, key renewal, key revocation and key escrow prevention. In particular, we present the first fully self-organized key revocation scheme for MANETs that does not require any trusted third party in the network. Our revocation scheme can be used to amend existing IBC solutions, be seamlessly integrated in our security framework and even be adopted to conventional public key solutions for MANETs. Our scheme is based on propagated accusations and once the number of received accusations against a node

reaches a defined threshold, the keys of the accused nodes are revoked. All communications are cryptographically protected, but unlike other proposed schemes, do not require computationally demanding digital signatures. Our scheme is the first that efficiently and securely enables nodes to revoke their own keys. Additionally, newly joining nodes can obtain previous accusations without performing computationally demanding operations such as verifying digital signatures. Several security and performance parameters make our scheme adjustable to the hostility of the MANET environment and the degree of resource constraints of network and devices. In our security analysis we show how security parameters can be selected to prevent attacks by colluding nodes and roaming adversaries.

In our proposed security framework, we utilize special properties of pairing-based keys to design an efficient and secure method for pairwise pre-authentication and a set of ID-based authenticated key exchange protocols. In addition, we present a format for ID-based public keys that, unlike other proposed formats, allows key renewal before the start of a new expiry interval. Finally, we are the first to discuss the inherent key escrow property of IBC schemes in the context of MANETs. Our analysis shows that some special features of MANETs significantly limit the escrow capabilities of key generation centers (KGCs). We propose a novel concept of spy nodes that can be utilized by KGCs to increase their escrow capabilities and analyze the probabilities of successful escrow attacks with and without spy nodes.

In summary, we present a complete authentication and key exchange framework that is tailored for MANET applications that have previously lacked such security solutions. Our solutions can be implemented using any pairing-based IBC scheme. The component design allows for the implementation of single schemes to amend existing solutions that do not provide certain functionalities. The introduction of several security and performance parameters make our solutions adjustable to different levels of resource constraints and security needs. In addition, we present extensions that make our solutions suitable for applications with sporadic infrastructure access as envisioned in the near future.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Guang Gong for her patient guidance, advice, support and encouragement throughout the course of this work. I would like to thank Prof. Lein Harn at the University of Missouri, Kansas City, for serving as my external examiner—I appreciate his valuable comments. I also want to thank Prof. M. Anwar Hasan, Prof. Douglas R. Stinson and Prof. Paul A.S. Ward at the University of Waterloo for serving as my committee and helping me to improve the quality of my thesis by their comments. It is a privilege to have such a great committee.

I would like to thank the Computer Security Division at the National Institute of Standards and Technology (NIST), Maryland, U.S.A. for giving me the opportunity to work in their group while completing my PhD thesis. I especially want to thank Dr. Lidong Chen at NIST for her kind support.

There are many other people who helped me in one way or another to complete this thesis. I want to thank my lab colleagues Yassir Nawaz, Dr. Nam Yul Yu, and Xinxin Fang for their support. I am extremely thankful to Dr. Antonio Izquierdo, Dr. John Kelsey, Dr. Mark Reitsma and Dr. Allen Roginsky at NIST for helping me to improve my thesis by their valuable comments and suggestions. Special thanks to Prof. Paul A.S. Ward for bringing wireless mesh networks to my attention and several fruitful discussions. Thanks to Prof. Urs Hengartner and Prof. Alfred J. Menezes for encouragement when I needed it.

I am extremely thankful to my friends who make my life a happy place. Special thanks to Dr. Frank Hamme who knows how to cheer me up more than anybody else does, Monika Juda for reminding me that "girls just want to have fun", Bianca Sudhues and Imke Koch for always being close to me despite the geographic distance, Solmaz Ghaznavi for making my rushed move from Canada to the US happen, and Catherine Donnelly for her tremendous support in my fight against the bureaucracy in various countries. I thank Krishna Bellamkonda, Luis Felipe Armenta and Sara Sadri for numerous refreshing coffee breaks and Tim Hortons for continues coffee supply. I would like to thank Lana Good at SOS Physiotherapy and my Yoga teachers for getting me back into shape after spending too much time in front of

the computer.

Last but not least I would like to thank my family for their unconditional love. I am very grateful to have parents who believe in me no matter what I do and I cannot thank them enough for their support throughout these years.

Thank you all!

Contents

List of Tables

List of Figures

List of Abbreviations

AAA	Authentication, Authorization, Accounting
AKE	Authenticated Key Exchange [protocol]
AODV	Ad hoc On-Demand Distance Vector [routing]
AP	Access Point
AS	Authentication Server
BDHP	Bilinear Diffie-Hellman Problem
BF	Boneh-Franklin [scheme]
BS	Base Station
CA	Certificate Authority
CRL	Certificate Revocation List
DARPA	Defense Advanced Research Projects Agency
DH	Diffie-Hellman
DLP	Discrete Logarithm Problem
DoS	Denial of Service
DSA	Digital Signature Algorithm
DSDV	Destination Sequence Distance Vector [routing]
DSR	Dynamic Source Routing
EAP	Extensible Authentication Protocol
ECC	Elliptic Curve Cryptography
EC-DH	Elliptic Curve Diffie-Hellman
GPS	Global Positioning System
GSM	Global System for Mobile communications
KCI	Key Compromise Impersonation [attack]
KDC	Key Distribution Center
KDF	Key Derivation Function
KGC	Key Generation Center
KRL	Key Revocation List

ID	IDentity
IBC	Identity-Based Cryptographic [scheme]
IBE	Identity-Based Encryption [scheme]
IDS	Intrusion Detection Scheme
LAN	Local Area Network
LFSR	Linear Feedback Shift Register
MAC	Message Authentication Code
MAN	Metropolitan Area Network
MANET	Mobile Ad hoc NETwork
MR	Mesh Router
OCSP	Online Certificate Status Protocols
PAKE	Password-Authenticated Key Exchange [protocol]
PAN	Personal Area Network
PAM	Pre-Authentication Model
PDA	Personal Digital Assistant
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PFS	Perfect Forward Secrecy
PR	Packet Radio
RFID	Radio-Frequency IDentification
RS	Revocation Server
SURAN	Survivable Adaptive Network
TTP	Trusted Third Party
UKS	Unknown Key-Share [attack]
WLAN	Wireless LAN
WMAN	Wireless MAN
WMN	Wireless Mesh Network
WMBB	Wireless Mesh BackBone
WPAN	Wireless PAN
WSN	Wireless Sensor Network

Chapter 1

Introduction

Over the past decade or so, there has been rapid growth in wireless and mobile applications technologies. Falling prices for personal mobile devices (such as cellular phones, personal data assistants (PDAs) and laptops) and an increasing number of available wireless services (such as Internet access via so-called "hotspots" at numerous public locations) have meant that wireless mobile communications have become an important part of our daily life. Furthermore, wireless links have replaced cords on our desks and ethernet cables in our offices. Exchanging sensitive information and accessing paid services over unprotected wireless links with unidentified and untrusted endpoints demand the deployment of security in wireless mobile applications. While security solutions and standards already exist for infrastructure based wireless networks—such as the widely deployed IEEE 802.11 standard [68] for wireless local area networks (WLANs) and IEEE 802.16 [72] for broadband WLANs—solutions for infrastructureless wireless mobile networks are still in their infancy, with many security problems unsolved. The latter type of network is commonly referred to as *mobile ad hoc network (MANET)*. As with every new technology, MANET applications introduce new security challenges that require the design of specialized security solutions.

This thesis is concerned with the design and analysis of security solutions for MANETs. We identify pairwise authentication and key exchange as primary security goals for securing communications in MANETs. Henceforth, we focus on

solutions achieving authentication and key exchange as well as all necessary prerequisites and related functionalities of these security goals. In the following section, we highlight some of the numerous MANET applications, give a motivation for securing MANETs, and outline the unique challenges of implementing security in these systems. In Section 1.2, we summarize previous work on MANETs. In the last section, we give an overview of the organization of this thesis.

1.1 Motivation

MANETs are infrastructureless wireless networks solely consisting of mobile nodes. Consequently, all nodes in a MANET must be capable of forming and maintaining the network by themselves; i.e., without the aid of an external central entity, predeployed infrastructure or backbone access. In addition, mobile nodes must carry out all network functions including routing. The described property is one of the main characteristics of MANETs and is often referred to as a *self-organization* property. Self-organization, combined with other MANET properties, allows MANETs to be instantly formed in a cost-efficient manner.

The unique features of MANETs enable numerous applications in a wide range of application areas, including military, government, health services and civilian applications. Initially, MANETs were studied and explored for military applications, such as for establishing instant communication infrastructures during rescue missions in war zones and disaster-affected areas [125], collecting data in hostile environments [116], and self-healing mine fields [67]. Initial investigations in the military sector lead to suggestions for the deployment of MANETs for countless other applications, such as law enforcement [125], virtual classrooms [125], connecting and reading out medical devices in hospitals [115], smart homes [115], wireless personal area networks (WPANs) [15], sharing resources [15], ubiquitous Internet access [7], instant networks for conferences and meetings [109], network games [7], and many others. A more extensive list of applications can be found in our survey [52].

While the need for security is apparent for highly security-sensitive military and

health service applications, we argue that communications in any kind of MANET should be protected. This is because MANETs are generally susceptible to various attacks because of use of wireless communication links. Wireless communication channels do not provide any physical protection and attacks on these channels are easy to carry out because they do not require expensive equipment, physical access or close proximity to the network. Attacks include passive eavesdropping and active attacks, such as modifying, fabricating, replaying, relaying messages as well as impersonation attacks. Another reason why MANETs require special protection is the weak physical protection and easy accessibility of mobile nodes, which makes them susceptible to compromise.

To prevent any kind of malicious modifications, messages should be integrity protected, which is typically achieved by applying cryptographic primitives such as hash functions [38] and message authentication codes (MACs) [39]. In order to thwart eavesdropping all confidential messages should be encrypted [89]. Both, integrity protection as well as encryption require cryptographic keys. As opposed to using static long-term keys, fresh cryptographic keys should be used to limit the amount of available ciphertexts in a crypto analysis as well as to reduce the damage of key compromise [89]. Such fresh session keys can be derived by executing key exchange protocols [18]. The described security properties are rendered useless in most applications if the authenticity of the communication ends cannot be ensured; i.e., nodes do not know who is sending encrypted and integrity protected data and who they share a session key with. To provide authentication, and thus thwart impersonation attacks, nodes can execute authentication protocols [18]. However, prior to the execution of authentication protocols, all nodes need to share some authentic credentials to be able to prove their identity to each other. In addition, if the establishment of a session key is desired, the authentic credentials need to contain some key material, such as public or secret keys. We refer to the initial exchange of credentials as *pre-authentication*. Upon pre-authentication, nodes can use their pre-shared key material to establish a secure channel. In addition to the described attacks that apply to any wireless network, new attacks arise from some of the unique features of MANETs; e.g., attacks on the multi-hop routing proto-

cols. Thwarting these attacks may also require the use of cryptographic keys to provide integrity, message authentication, confidentiality, and/or entity authentication. For example, many secure routing protocols require shared keys between next hop neighbors or source and destination nodes [63, 65, 100].

We conclude that pre-authentication, authentication and key exchange are primary security goals in MANETs because once provided, all other security properties can be achieved in a straight-forward manner. An overview of security goals and applied security mechanisms used throughout this thesis is in Section 2.2. We can observe that the identified primary security goals are the same as in other infrastructure-based wired or wireless networks. However, existing security solutions for such networks cannot simply be adopted to secure MANETs because of the unique properties of MANETs. Basically, the very same features that enable exciting new applications are the same properties that make implementing security a very challenging task. The main challenges that need to be addressed when designing security solutions for MANETs are:

1. Lack of infrastructure
2. Resource-constrained nodes and communication links
3. Node mobility and network dynamics
4. Likely node compromise

The most challenging property is the lack of infrastructure in MANETs; i.e., the self-organization property. This affects the choice and design of the pre-authentication mechanisms, authentication and key exchange protocols as well as all prerequisites and miscellaneous mechanisms concerning the key management. Establishing a secure pre-authentication channel is a challenging prospect in MANETs because nodes need to pre-share keys over insecure channels without the aid of a trusted third party (TTP). Lack of infrastructure also causes problems in most authentication and key exchange protocols. For example, in public key infrastructure (PKI) solutions, the network must provide a way to issue and distribute public key certificates which is typically done by a central certificate authority (CA). In addition,

PKI solutions must address the problem of providing certificate revocation lists (CRLs) or other forms of revocation status checks without having a central server offering this kind of information. On the other hand, symmetric key solutions which distribute secret keys using a Kerberos server [97] or other type of TTP are not applicable in MANETs either.

The second challenge, i.e., resource constraints, requires solutions to be efficient with respect to computational and communication costs. Mobility and dynamics do not directly affect security protocols; however, these properties must be considered for designing suitable key management mechanisms. Due to weak physical protection and easy accessibility, node compromise and thus compromise of key material, is likely. Hence, security solutions should be sufficiently resilient to the compromise of some nodes and minimize the damage of such compromises if and when they occur. We conclude that securing MANETs requires the design of new security solutions that address all unique features and challenges of MANETs.

1.2 Previous Work

In this section, we review some previous work on MANETs. Historically, MANET research was driven by the military to enable multi-hop communications in networks that are easy, fast and inexpensive to set up and do not require any infrastructure. The next step was developing efficient multi-hop routing protocols to replace the previous flooding approach. Once basic functionalities were achieved, researchers started to work on security solutions for MANETs. In the remainder of this section, we give a brief overview of the history of MANETs, routing protocols and security solutions.

1.2.1 Early MANET Projects

The initial research on MANETs was driven by the military, more specifically by the Defense Advanced Research Projects Agency (DARPA). The first class of MANETs were so called packet radio (PR) networks [76] developed under the sponsorship of DARPA in the late seventies. In PR networks, nodes broadcast messages to their

in-range neighbors, which in turn relay the received messages to their neighbors, thus establishing a multi-hop ad hoc network. The used radios in PR networks were expensive, heavy, had slow CPUs and required a lot of energy. The next generation of MANETs sponsored by DARPA in the eighties used smaller, less expensive and lower power radios in so-called Survivable Adaptive Radio Networks (SURAN) [37]. The projects that followed focused mainly on the miniaturization of devices as well as connectivity of a larger variety of devices (e.g. Piconet [12] and SmartDust [113], respectively).

1.2.2 MANET Routing

The next wave of research projects focused on more efficient and sophisticated multi-hop routing protocols for MANETs, replacing the broadcast approach of previous systems. Due to the dynamic network behavior and lack of static routers in MANETs, new protocols needed to be designed because existing routing protocols for LANs and WLANs are not applicable. In 1994, the Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) protocol [117] for MANETs was proposed, followed by the Dynamic Source Routing (DSR) protocol [75] in 1996 and the Ad Hoc on Demand Distance Vector (AODV) routing protocol [118] in 1997. Today most MANET implementations employ DSR, AODV or improvements of one of these two routing protocols.

After security issues of MANETs were brought to researchers' attention (see next Section), numerous attacks exploiting the special properties of multi-hop routing protocols were discovered and addressed [5, 21, 22, 63, 64, 65, 66, 88, 100, 110, 122]. Most attacks on routing protocols are executed by malicious nodes and are aimed to disrupt the network connectivity, e.g. blackhole [122], wormhole [64] and rushing attacks [66]. The routing functionality may also be disrupted by network nodes that refuse to forward messages in order to save their own battery power. These kind of nodes are referred to as *selfish nodes*.

Some of the proposed secure routing protocols employ symmetric cryptography [63, 65, 100] or public key cryptography [5, 110, 122]. However, some attacks cannot be addressed solely by cryptographic means. For example, rushing attacks

can be prevented by accepting a random—as opposed to the first—routing request [66], whereas wormhole attacks can be addressed by using timestamps or location information as part of routing packets [64], and selfish nodes by rewarding nodes that faithfully forward messages (as well as punishing selfish nodes [22]). A more general approach to detect and exclude maliciously acting network nodes is to use monitoring schemes in which nodes observe their neighborhood for any malicious behavior [21, 88].

1.2.3 MANET Security

Finally, in 1999, the importance of securing MANETs including their special security needs and challenges were discussed for the first time [115]. This groundbreaking paper triggered an explosion of research in MANET security. Due to the large number of published papers, we organize the security review in terms of utilized cryptographic primitives rather than chronologically. We refer to Chapter 3 for a detailed discussion of advantages, disadvantages and applicability of some solutions and only outline the different lines of research here.

Due to their efficiency, symmetric crypto schemes seem well-suited to address the resource constraints of MANETs and their nodes. Once two nodes share a secret key, they can use this key to authenticate each other, establish fresh session keys or achieve other desirable security properties. However, the initial key distribution in symmetric schemes, i.e. the pre-authentication, poses a major problem because of the absence of an on-line key distribution center (KDC) in MANETs. Hence, existing symmetric key solutions for infrastructure-based networks are not applicable to MANETs, such as the Kerberos authentication system [97] or symmetric EAP (Extensible Authentication Protocol) methods—such as EAP-GPSK [30]—that use RADIUS or DIAMETER authentication servers. To enable the use of symmetric cryptography in MANETs, secret keys must be either pre-deployed or exchanged over a secure channel within the network. Several probabilistic key pre-distribution protocols, which assign each node a subset of the entire key pool, have been proposed [36, 84]. In these schemes, two nodes wishing to communicate check if they share a secret key. If they do not, the nodes try to find a common neighbor with

whom they both share a key [36]. In [84] it is assumed that nodes' locations are static and known before deployment. This additional information is used to increase the probability that two neighboring nodes share a key. Probabilistic schemes limit the damage of key compromise and save memory space compared to network-wide secrets and pairwise shared keys among all nodes, respectively.

If key pre-distribution is not feasible, nodes must establish shared keys in the network. In the first proposed symmetric scheme for MANETs [115], secret keys are exchanged by physical contact, which ensures a confidential and authentic key exchange. Another symmetric scheme for MANETs suggests sharing a low-entropy password among participating nodes/users [4], for instance by writing it on a blackboard in a conference room, and then running a multi-party password-authenticated key exchange (PAKE) protocol to derive a cryptographically strong key. If only efficient (unidirectional) authentication is required without the need for establishing keys, the use of Lamport's hash chains [80] has been suggested for MANETs [120, 121]. Here the anchors of the hash chains must be securely exchanged, where [120] uses previous experiences with a node and [121] digitally signed anchors for this purpose. Note that the latter compromises the efficiency of a solely symmetric solution. The current IEEE standards for WLANs IEEE 802.11 [68] as well as Wireless Personal Area Networks (WPANs) IEEE 802.15 [70] and IEEE 802.15.4 [71] establish shared keys by pre-loading the keys into all devices. However, security amendments IEEE 802.11i [69] and IEEE 802.16e [73] both use the EAP framework for authentication and key establishment which requires infrastructure access and are thus not suitable for MANETs.

The limitations of symmetric key solutions caused by the key distribution problem in MANETs triggered the research on public key solutions. The first papers on public key solutions focused on the implementation of an on-line CA that issues and distributes public key certificates within the network in a self-organized manner [67, 85, 125]. In [85, 125], the power and tasks of a CA are distributed to several network nodes using a (k, n)-threshold scheme. In [67], trust is generated in a PGP manner and each node issues and distributes certificates. To avoid the complications of implementing an on-line CA, some papers proposed exchanging public

keys over location-limited—and thus authentic—channels which makes the use of public key certificates redundant [7, 23]. Sometimes additional information such as the geographic location of a node is used to establish an authentic channel [23].

Due to their efficient key management and other desirable properties, IBC schemes have been recently considered for securing MANETs [33, 77, 124]. Solutions proposed in [33, 77] both use an internal key generation center (KGC). The KGC is emulated using a (k, n)-threshold scheme, as has been previously proposed for internal CAs in PKIs. The authors claim that their schemes are more efficient than fully self-organized PKIs because of the efficient key management of the underlying IBC schemes. The authors of [124] propose an IBC scheme in which nodes are initialized by an external KGC and all other tasks such as key renewal and key revocation are executed in the network. This approach provides a more efficient network set up than the solutions in [33, 77] but compromises the self-organization property during this phase.

Key revocation and key renewal are essential mechanisms in all public key based security solutions for MANETs. However, most proposed solutions either do not provide such mechanisms at all, or only outline a solution. For instance, [33, 67, 77] do not provide any mechanism for key revocation, whereas [125] suggests that the internal CA should be able to revoke collaboratively certificates, but does not introduce any algorithm. In [32, 85, 124], so-called *accusation schemes* are used where each node is able to accuse other nodes of malicious behavior. If the number of accusations is greater than a certain threshold δ, the certificate is considered to be revoked. The revocation scheme in [85] is outlined in one paragraph with a suggestion to implement a sign&broadcast approach to securely distribute the accusation tables. A more sophisticated scheme is introduced in [32]; however, the propagation of accusation tables in this scheme is not secured and nodes derive their own accusation tables by finding majorities over received accusations. Key revocation and key renewal mechanisms for IBC schemes are introduced [124] in which nodes send their accusations to fixed assigned entities that are part of a distributed internal KGC.

1.3 Thesis Overview

The diversity of MANET applications prevents the design of a universal one-size-fits-all solution. Existing security solutions for MANETs, as reviewed in the previous section, are only suitable for some selected MANET applications. In this thesis we specify some common parameters of targeted MANET applications and present solutions for these MANETs. We keep the specifications as general as possible and show how our proposed solutions can be adopted to accommodate other classes of MANET applications. Hence, our solutions are applicable to a large number of MANET applications that have no sufficient security solutions yet. In addition, we discuss how the performance and security of our solutions can be further improved by taking advantage of (sporadic) infrastructure access envisioned for the next generation of MANETs.

Many security goals are completely neglected in existing security solutions for MANETs or are treated in an insufficient manner. For instance, pre-authentication has been widely ignored. We identify pre-authentication as crucial to achieving other security goals and discuss several pre-authentication models including solutions in each of the model. We believe that IBC schemes have some distinctive features that make them an excellent tool for MANET security. Until now, the role of IBC schemes as enabler for security in MANETs has not been thoroughly explored. In this thesis we present the first complete ID-based authentication and key exchange framework for MANETs. Our solutions improve existing schemes and provide solutions to problems that have not been addressed before. For instance, we present the first fully self-organized revocation scheme for IBC schemes deployed in MANETs that does not require any external or internal KGC. Our scheme is the first revocation scheme for MANETs that provides an algorithm that allows nodes to securely and efficiently revoke their own keys. Furthermore, our revocation scheme is the first that allows newly joining nodes to receive previous accusations without the need of verifying signatures which makes our scheme efficient. Several performance and security parameters allow our solution to be adjusted to the level of hostility and constraints posed by particular MANET applications. We show in our security analysis how parameters should be selected to prevent attacks

by colluding and roaming adversaries.

In this thesis, we introduce the first key renewal algorithm for ID-based keys that allows key renewal before the next expiry interval starts. Furthermore, we are the first to address the key escrow property inherit in all IBC schemes in the context of MANETs. We propose the novel concept of so-called spy nodes that may be deployed by KGCs to increase their key escrow abilities. We then analyze the probability of key escrow attacks with and without spy nodes and present countermeasures which can significantly reduce the likelihood of key escrow in MANETs. In addition, we are the first to discuss the suitability of ID-based AKE protocols in MANETs and present a set of ID-based AKE protocols targeted to the computational and communication constraints of MANETs. The provided security-performance analysis allows the selection of the best-suited protocol with respect to the degree of constraints and required security level of particular MANET applications. In summary, we provide the first complete ID-based authentication and key exchange solution for MANETs.

The remainder of this thesis is organized as follows. In Chapter 2, we introduce definitions and notations used throughout this thesis. In Chapter 3, we identify and categorize secret and public key pre-authentication models for MANETs and discuss their applicability. In Chapter 4, we first discuss features and challenges of IBC schemes employed in MANETs. Then we introduce an ID-based authentication and key exchange framework consisting of algorithms for system set up, key derivation, key distribution, and pre-authentication. In addition, we present and analyze a set of ID-based AKE protocols. In Chapter 5, we introduce a fully self-organized key revocation scheme as well as a key renewal scheme for MANETs. In Chapter 6, we analyze key escrow in the special context of MANETs and propose the novel concept of spy nodes. In Chapter 7, we analyze opportunities and challenges of envisioned future applications of MANETs. In particular, we modify the key revocation and key renewal schemes from Chapter 5 and introduce more AKE protocols such that they take advantage of sporadic network access in mesh networks. Finally, we summarize our contributions as presented in this thesis and discuss directions for future work in Chapter 8.

Chapter 2

Preliminaries

For an easier understanding of the presented solutions and results in this thesis, we briefly review necessary background information and concepts about MANETs and security primitives. Furthermore we define terms and notations that are used throughout this thesis. Please refer to Table 2.3 for a list of used symbols.

2.1 MANETs

The diversity of MANET applications and research projects, as outlined in Sections 1.1 and 1.2, respectively, use or emphasize different unique properties of MANETs, respectively. Hence, clear definitions of MANET properties and parameters are missing. Henceforth, we distinguish between *properties* and *parameters*, where MANET properties are universal, whereas parameters depend on particular applications or implementation environments. In this section, we present a definition for MANETs that we use in the remainder of this thesis. Since this thesis is concerned with security aspects in MANETs, we limit our focus to unique MANET properties and parameters that make implementing security a challenging task. We specify parameters of target MANET applications and limit our discussions to MANET applications that do not have sufficient security solutions yet.

2.1.1 Wireless Communication Technologies

Typical wireless communication technologies used in MANETs are IEEE 802.11 [68], IEEE 802.15.1 [70]/Bluetooth [15], IEEE 802.15.4 [71]/ZigBee [126], IEEE 802.16 [72], and IrDA infrared data protocols [74]. For an easier comparison, we list the communication ranges, data throughput, and quantitative power consumptions of these communication technologies in Table 2.1. Note that except IrDa, all listed standards use radio technologies. If two nodes i and j in a MANET are in each others immediate transmission range, they can directly exchange messages with each other. This is sometimes called one-hop communication and illustrated in Figure 2.1-a. However, from Table 2.1 we can observe that the aforementioned communication technologies have a very restrictive communication ranges. Hence, to enable nodes to communicate with other nodes outside their communication ranges, multi-hop routing must be used. Therefore, each network node acts as router r and packets are repeatedly forwarded to other nodes in direct communication range, until the packets reach their destinations. Multi-hop routing between nodes i and j via intermediate nodes r_1, r_2, r_3, and r_4 acting as routers is illustrated in Figure 2.1-b, where, for simplicity, we assume that transmission range Tx and reception range Rx are equal, i.e. $R = Tx = Rx$, and the same for all network nodes. Another constraint imposed by the employed wireless technology are the limited bandwidths (see Table 2.1).

2.1.2 Node Properties

Typical MANET nodes are laptops, PDAs, pocket PCs, cellular phones or any other mobile wireless devices. In contrast to WLANs, MANETs typically consists of a set of similarly constrained devices because no servers, routers or other powerful entities are deployed or accessible. Due to their mobility, MANET devices are typically lightweight and battery-operated. Furthermore, MANET devices are generally inexpensive to enable wide usage. These features of MANET devices lead to several resource constraints, namely:

- small CPU

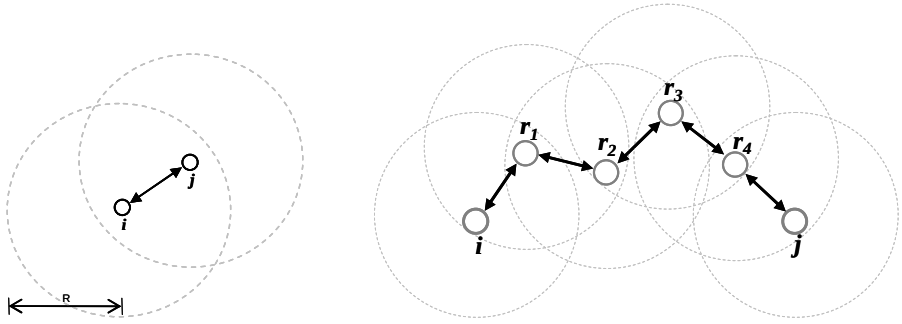
	IEEE 802.11 [68]	IEEE 802.15.1 [70] /Bluetooth [15]	IEEE 802.15.4 [71] /ZigBee [126]	IEEE 802.16 [72]	IrDA [74]
Range	∼ 100m	∼ 10-100m	∼ 10m	∼ 6-8 km	∼ 1m
Data throughput	∼ 2-11 Mbps	∼ 1 Mbps	∼ 0.25 Mbps	∼ 45 Mbps per channel	∼ 100 Mbps
Power consumption	medium	low	ultra low	high	medium

Table 2.1: Ranges, Bandwidths and Power Consumption of Some Common Wireless Communication Technologies

- small memory
- limited battery power
- weak physical protection

These constraints, as first summarized in [115], severely limit the computational and communication capabilities of MANET devices and thus of the overall network. To illustrate some resource constraints, we list the technical specifications of several representative MANET devices in Table 2.2. Note that the communication ranges of MANET devices vary depending on the supported wireless technology. For example, laptops and PDA typically support IEEE 802.11 and 802.15.1 standards and thus have communication ranges from 10-100m, whereas sensors and RFID have far more limited ranges, typically between a few centimeters to several meters.

Due to the limited CPU power, some computationally intensive operations may not be feasible on a MANET device. Even if such operations are feasible, the number of executions should be limited because of the power constraints of the devices. In addition, complex programs or large sets of system data may require too much memory space. In general, CPU and storage technology are advancing fast.



(a) (b)

Figure 2.1: Communication between nodes i and j: (a) one-hop communication, (b) multi-hop communication.

Moore's law states that transistor density doubles every two years [93] resulting in faster small size CPUs, and Shugart's law states that magnetic storage prices per bit halve every 18 months. Furthermore, bandwidth constraints will be less stringent over time due to Gilder's law [42] that states that bandwidth grows at least three times faster than computing power. However, technological advances for extended battery life is comparably much slower. In addition, batteries might not be rechargeable in the network and a node is excluded from the network once its battery is drained. From the previous discussion we conclude that power is by far the most scarce resource in MANETs and thus conserving energy is crucial. Power can be conserved by limiting the number of transmitted and received packets as well as the number and kind of executed computations. Please note that transmitting packets is the most power consuming operation.

Finally, we observe that node compromises are likely because nodes offer only weak physical protection and are easily accessible by users or other malicious parties.

2.1.3 Network Properties

After discussing the properties of employed communication technologies and MANET devices, we are ready to define and summarize the properties of MANETs.

	Laptops	Pocket PCs/PDAs	Smart Phones	Sensors	RFID
CPU	Intel Core 2 Duo, 2GHz	Intel XScale PXA270, 520MHz	Texas Instrument OMAP850, 200MHz	MICA Mote, Atmega 128L, 4MHz	none
Battery	High Capacity Ion Battery, 10W-120W	Ion Battery: 100mW-10W	Ion Battery: 100mW-10W	2 AA batteries	no battery (passive), powered by reader
Memory	$\sim$ 60-200 HD, 2-4 GB RAM	64MB RAM, 192 ROM	64 MB RAM, 128 MB ROM	128KB Flash memory, 4KB RAM, 4KB ROM	128bytes to several KB ROM

Table 2.2: Technical Specifications of Some Representative MANET Devices

Def. *MANETs*: Wireless networks that are spontaneously formed by a group of mobile nodes without the help of any infrastructure. Once the network is formed all tasks are performed in a self-organized fashion. Nodes communicate over short-range wireless links, where each node acts as router enabling multi-hop routing to increase the nodes' communication ranges.

From the above definition we observe the following properties of MANETs:

1. infrastructureless

2. transient

3. dynamic
4. wireless
5. multi-hop
6. mobile network nodes
7. resource constrained network nodes (power, CPU, memory)
8. similarly constrained network nodes

The first property refers to the lack of infrastructure in MANETs. This follows that network nodes must carry out all network operations which is referred to as self-organization property. Note that the self-organization property may be relaxed during certain network phases or in some applications as we discuss in Sections 2.1.4-2.1.6. MANETs are transient because they are spontaneously formed for a specific purpose or to offer a certain service. Networks are dynamic because nodes may join or leave the network at any time. As mentioned in Sections 2.1.1 and 2.1.2, nodes are mobile, resource constrained, communicate over short-range wireless channels, use multi-hop routing to increase their communication ranges, and are similarly constrained.

Sometimes wireless sensor networks (WSNs) are considered as type of MANET. However, in this thesis we clearly distinguish between WSNs and MANETs. Both types of networks have different application areas and many distinguishing properties. Typically sensors are deployed in an area where they establish a network and start collecting and processing data, e.g. for monitoring the environment. We briefly list the most distinguishing properties of WSNs compared to MANETs in the following:

- number of sensors in network can be orders of magnitude larger that nodes in a MANET
- broadcast communications versus point-to-point
- many-to-base station communications versus one-to-one

- severely constrained sensors versus moderately constrained devices (see Table 2.2)
- typically stationary sensors versus mobile devices

The different application areas and properties of MANETs and WSNs lead to completely different security goals and challenges for designing suitable security solutions. In this thesis we focus on solutions for MANETs and all presented solutions are designed for securing MANETs and not targeted at WSNs.

2.1.4 Network Phases

We distinguish two network phases in MANETs:

1. network initialization phase
2. running network phase

During the first phase, all nodes that are present at the time of the network formation are initialized. The initialization is performed by a TTP and includes the distribution of system parameters and cryptographic key material to each node. Trusted third parties can be central external entities or distributed internal entities consisting of a group of network nodes. We discuss external and internal TTPs in more detail in the next section. Basically, all parameters needed to execute network protocols, including security protocols, must be distributed to all present nodes during the network initialization. Note that the distribution of secret key material requires authentic and confidential channels, whereas the distribution of public key material requires authentic channels.

Upon network initialization, new nodes may join or present nodes leave the network at any time in the running network phase. Note that new nodes that join the network after the network initialization phase must also obtain system parameters and cryptographic key material. The node initializations may occur outside the network, i.e., before the nodes join the network, or within the network by other network nodes. During the running network phase, two or more nodes may

execute network protocols utilizing parameters and keys that have been established during network or node initialization.

2.1.5 Trusted Third Parties (TTPs)

We differentiate between external and internal TTPs as well as central and distributed TTPs. An *external TTP* is an entity outside the network that is trusted by all network nodes. The TTP can consist of a single central entity or n distributed entities. Latter implementation is sometimes chosen to increase the overall trust in the TTP because trust can be maintained even if some of the TTPs may not be trustworthy. An *internal TTP* is a distributed TTP consisting of n network nodes, where $n \leq \Omega$ with Ω being the total number of nodes in the network. Here, power and capabilities of a TTP are distributed to n network nodes. Distribution of power is desirable to avoid single point of failures. This is necessary in MANETs because of the likelihood of node compromises. So-called (k, n)-threshold schemes [111] can be used to implement distributed external or internal TTPs. In that case any k out of n distributed TTPs can collaboratively execute some tasks, such as decrypting or signing messages. There are two possible cases of distributed internal TTPs: (1) *distributed TTP with special nodes* and (2) *distributed TTP with conventional network nodes*. In the first case, n special nodes that are more powerful than the other $\Omega - n$ network nodes, e.g. in terms of computational and battery power, represent the TTP. These special nodes have been initialized during the network initialization phase. This approach has been used in [125]. However,it contradicts Property 8 in our list of MANET properties, namely the assumption that MANET nodes are similarly constrained. In the second model, any group of n network nodes can be selected to represent the TTP. This approach has been used in [33, 77, 78, 85]. Please note that while enabling more features and flexibility, distributed on-line TTPs always impose a lot additional computational and communication costs to the network due to the use of threshold schemes.

A TTP can have several roles in a network, for instance, a TTP could initialize nodes with necessary key material during network initialization or before joining the network, distribute session keys to nodes that wish to securely communicate

with each other in the running network, or help to verify public key certificates by providing CRLs.

2.1.6 Parameters

Unlike MANET properties which are universally valid, there are a number of MANET parameters that may vary for different applications. Parameters have an impact on the protocol design for particular applications. In this section, we summarize some typical MANET parameters.

Pre-Existing Trust: In some applications, network nodes have existing trust relationships with each other, e.g., based on personal relationships, mutually trusted nodes, previous experience with some offered services, or reputation. These relationships may exist prior to the network initialization or are established over time in the running network. In other applications there may not be any pre-existing trust among nodes. However, we always assume that all network nodes trust the external or internal TTP in the network.

Pre-Authentication Channel: Another crucial parameter is the available channel for the initial credential exchange between pairs of nodes, i.e. the pairwise pre-authentication. Pre-authentication may implicitly occur during network initialization phase, e.g. the TTP distributes one secret key or a set of pairwise secret keys to all nodes. In that case, the pre-authentication channel must be established between each node and the TTP, where the channel must be authentic and confidential. Hence, no additional pre-authentication channel between pairs of nodes is necessary. If no pre-shared secrets exist, the pre-authentication must be carried out between pair of nodes in the running network. In that case, the pre-authentication channel must be authentic to exchange public keys or authentic and confidential to exchange symmetric keys. However, establishing secure pre-authentication channels without sharing any credentials is difficult. Different solutions of how such pre-authentication channels can be established are discussed in Chapter 3.

Network Size: MANETs can significantly vary in size, i.e. the number of network

nodes Ω, depending on the application. In addition, the network size may change over time due to joining or leaving nodes. The size of a network is crucial for the design of security solutions, e.g. large or very dynamic networks require scalable solutions, whereas small networks may allow simplified solutions.

Hierarchical vs. Flat Topology: Hierarchical MANETs have been proposed as alternative to flat topologies to overcome some limitations of the latter, as for instance described in [16]. Hierarchical MANETs have two or more network layers, each layer consisting of a set of similar powerful nodes. For instance, the lowest layer consists of the least powerful nodes and every level up consists of more powerful nodes, where the top level may have access to some infrastructure, such as a backbone network or the Internet. In this way, all computationally challenging operations can be shifted from constrained to more powerful nodes. Although this model is very attractive for designing security protocols, the applicability is limited to certain restrictive applications.

Controlled vs. Non-Controlled: In general, all nodes in a MANET have similar roles and are assumed to have similar resource constraints. However, some MANET applications might have a network node entity that controls other nodes. The controller might be a more powerful entity, whereas the other nodes are very constrained and simply execute orders. This concept was first introduced in the "Resurrecting Duckling Model" [115] in which one node acts as a controller (mother duck) and several devices (the ducklings) follow the controller's instructions. Potential applications for MANETs with controller nodes are industrial control and building automation [91]. Designing protocols for networks with controllers is easier, however, applications are limited and generally only apply to sensor networks.

One Domain vs. Multiple Domains: All devices in one domain share the same domain parameters, including security parameters of implemented cryptographic algorithms. For example, domain parameters could be shared keys that have been distributed during network initialization, the public key certificate of the domain's root CA or system parameters required for cryptographic computations. A domain has one central or distributed TTP that is trusted by all nodes in that domain.

The TTP is responsible to initialize all nodes with the domain parameters. In most sensor networks, it is reasonable to assume one domain, whereas applications with multiple domains can be envisioned for MANETs. Providing authentication in networks with several domains is harder to implement because nodes do not necessarily trust a TTP of another domain. However, for cross-domain security, either nodes need to trust the TTP of the other domain or there must be an agreement among the TTPs of different domains. Furthermore, nodes from different domains that wish to authenticate each other, have to first agree on some common system parameters. For instance, nodes from different domains typically do not pre-share any secret keys and public key certificates are issued by different domain CAs and thus cannot be easily verified.

Degree of Resource Constraints: The level of resource constraints depend on the used network devices and employed communication technology, as illustrated in Tables 2.1 and 2.2. This in turn is determined by the application. Depending on the degree of constraints some security solutions might be practical or not. For instance, computationally demanding operations such as modular exponentiations in public key algorithms might be infeasible on some devices. In addition, some protocols demanding a large number of computations might not be feasible due to limited battery power. Finally, protocols requiring the exchange of many messages might be impractical due to power and bandwidth restrictions. As mentioned earlier, the most scarce resource is power. Security solutions need to be designed according to the degree of resource constraints of particular MANET applications.

Location Awareness of Devices: In some scenarios, network nodes have special equipment that provide location information, such as geographical coordinates. Location information could be utilized to establish an authenticated channel between nodes [23]. For instance, some high-end PDAs are already equipped with GPS chips [48]. There are many systems that can be utilized to provide location coordinates, e.g. : (1) satellite navigation systems, such as GPS or the European equivalent Galileo [40]; (2) systems for locating devices inside a building using visual, ultra sonic, radio, or infrared channels; and (3) network based positioning

system, such as GSM [49] and WLANs [68]. Special equipment for tracking nodes is unnecessary if their location is predictable. For instance, in some MANETs, nodes may have an expected location which can be used in location-based key establishment [84].

Availability of Trusted Third Party (TTP): The availability of a TTP constitutes one of the major challenges of MANETs and is crucial for the design of security solutions. We distinguish four cases of availability, described in the following paragraphs and illustrated in Figure 2.2. The four rows in the figure correspond to the four availability cases, where the first column describes the network initialization phase, the second column the event of a joining node in the running network and the third column the event of two nodes executing a network protocol in the running network.

AV-1: External TTP always available

In this scenario an external TTP is accessible by all network nodes at any time, i.e. during network initialization, node initialization and during execution of network protocols. This scenario is generally not considered as an option in MANETs, because it requires the existence of an infrastructure and interferes with the self-organization property of MANETs. However, with the growing number of available network access points at various locations, it is reasonable to assume an Internet or backbone connection in some MANET applications. An already existing example of such MANETs are wireless mesh networks in which a MANET can be an extension of an existing infrastructure. When designing security solutions for this type of MANET applications, existing solutions from infrastructure-based networks (such as WLANs) can be modified to cope with the resource constraints of MANETs. On the other hand, solutions designed for MANETs without TTP access, can be optimized to take advantage of the infrastructure access.

AV-2: TTP available at network and individual node initializations

The second scenario comprises applications in which an external TTP is accessible by all nodes present at network initialization and all nodes that subsequently join the network. This assumption is not as restrictive as it might seem, because the TTP does not need to be accessible by all network nodes every time a new node

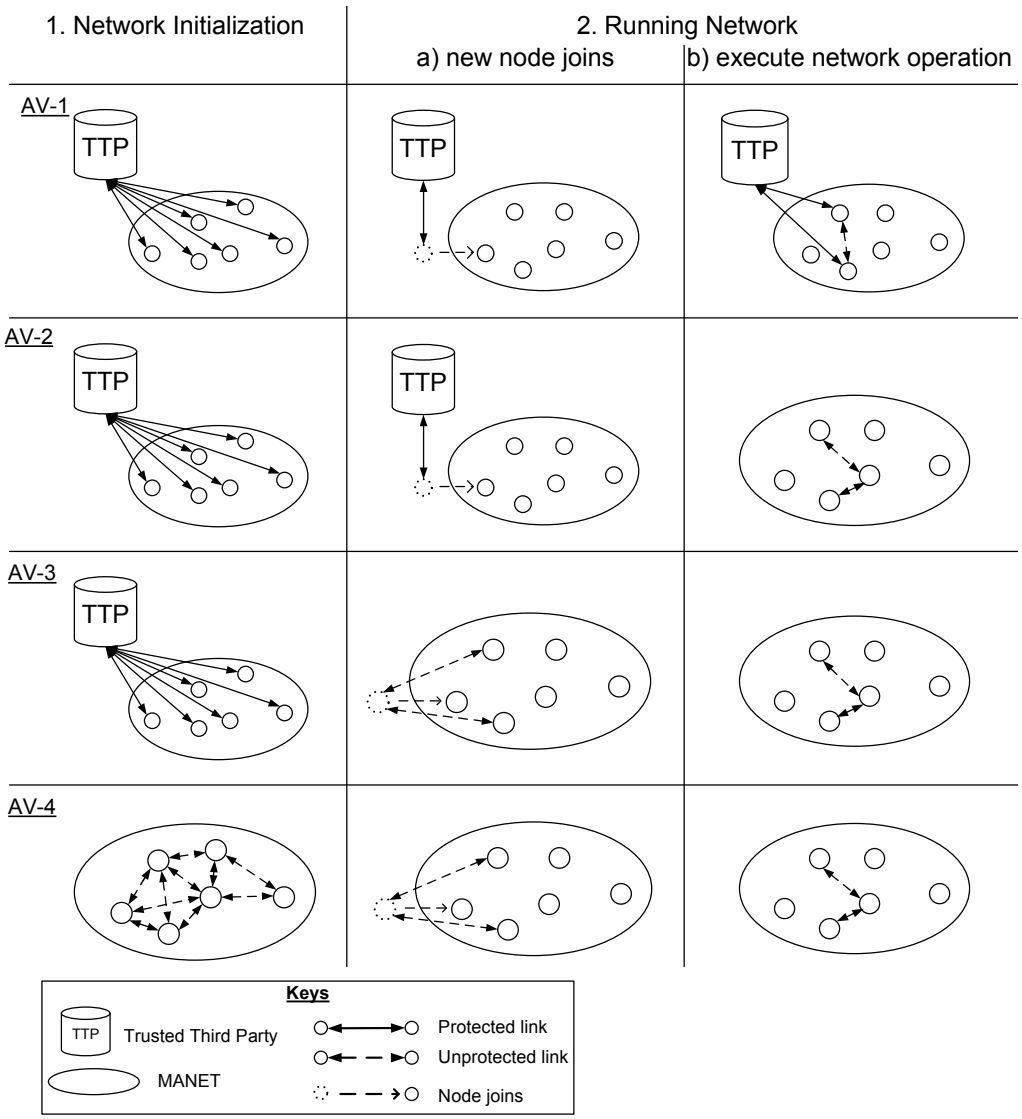


Figure 2.2: Scenarios of TTP Availabilities *AV-1* – *AV-4*: 1. during network initialization; 2. in the running network when a) new nodes join or b) two nodes execute a network operation.

joins, but only to the new nodes. Clearly, the external TTP does not need to be anywhere near the actual network. For instance, there might be applications in which nodes contact a TTP to receive the required system parameters and keys before joining the network. In these cases, the network itself is still self-organized and the present nodes have no access to a TTP.

AV-3: TTP available at network initialization phase

In this scenario only nodes that were present at the time of the network formation, i.e. during network initialization, are initialized by an external TTP. As a consequence, subsequently joining nodes need to be initialized within the network by other nodes. This is accomplished by a distributed internal TTP, i.e. a group of nodes that are already a part of the network. In addition, all protocols are executed without the help of any external TTP. Hence, the network is completely self-organized upon network initialization.

AV-4: No external TTP available

In the last scenario, no external TTP is available at any time. Hence, a distributed internal TTP consisting of network nodes needs to take over all TTP tasks during network initialization as well as in the running network. Consequently, network nodes themselves are responsible to carry out the initial network set up, initialize newly joining nodes and execute protocols. These type of applications require networks to be completely self-organized at any time and thus constitute the most challenging scenarios for security solution design. This scenario represents the original vision of MANETs, because absolutely no infrastructure is necessary. However, this scenario may be more restrictive than the actual environment of most MANET applications and thus may put an unnecessary burden on the protocol design.

2.1.7 Wireless Mesh Networks (WMNs)

Wireless mesh networks (WMNs) are a class of MANETs with at least sporadic infrastructure access, i.e. TTP availability AV-1 or AV-2 as described in the previous section. We discuss security solutions for WMNs and their differences to MANET solutions in Chapter 7. The number of deployed WMNs is growing and WMN components such as wireless mesh routers are commercially available to allow an easy

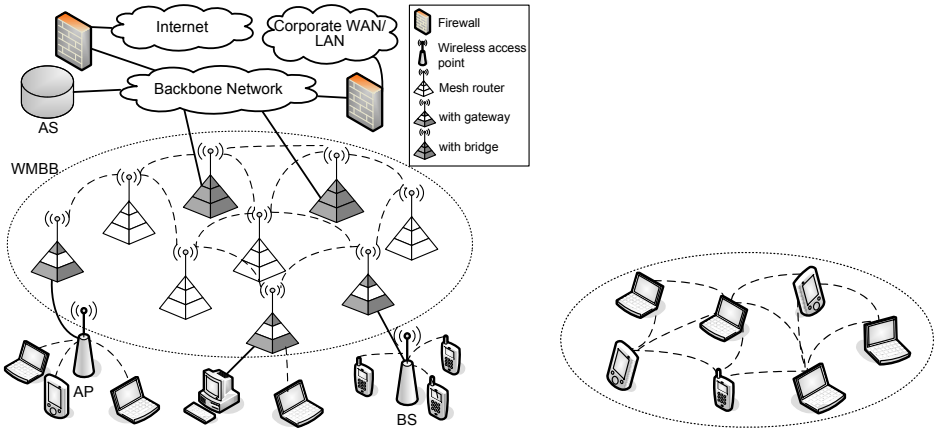


Figure 2.3: Types of WMNs: (a) Infrastructure WMN, (b) client WMN.

set up. WMNs are intended to integrate different kinds of existing wireless technologies, such as IEEE 802.11 WLANs, IEEE 802.16 broadband WMANs, IEEE 802.15 WPANs, cellular networks, etc.. It is envisioned that this integration allows mobile clients to connect to the Internet and use other web-based services from virtually everywhere.

An overview of a typical WMN architecture is illustrated in Figure 2.3-a. From the figure we can observe that mobile wireless clients can access a WMN as long as they are in communication range of a mesh router (MR), wireless access point (AP) or base station (BS). If mesh clients are not in range of any of these entities, they might be still able to access the network through a multi-hop connection consisting of other mesh clients, where the last hop is in range of one of the access points. Mesh clients typically communicate over wireless links, but may have wired connections to MRs. On the other hand, AP and BS operate one radio for client communications and share a wired connection to a MR. MRs offer bridge and gateway functionalities to allow inter-operability of various radio technologies and operate at least two radios, one to communicate with mobile clients, and another one to communicate with other MRs. MRs communicate in a multi-hop fashion to form the wireless mesh backbone (WMBB). MRs are inexpensive devices that are powered from a permanent source, e.g. AC mains powered, and some MRs are equipped with

gateway functionally to connect the WMBB to the Internet or other backbone networks. MRs are typically stationary and route messages from mobile clients to a MR with gateway to the desired network or network service. The backbone of the network typically includes some kind of authentication server (AS) that is used to verify the authentication and authorization of mobile clients that wish to access the network. Typically AAA (Authentication, Authorization, Accounting) servers are used for this purpose, e.g. RADIUS [105] or DIAMETER [106].

Three types of WMNs are commonly distinguished: (1) infrastructure or backbone WMNs (see Figure 2.3-a), (2) client WMNs (see Figure 2.3-b), and (3) hybrid WMNs. Infrastructure WMNs consist of wireless mobile clients which are in immediate communication range of access points, access points, the WMBB, and the backbone network. On the other hand, a client WMN solely consists of mobile clients that are all out of range of access points to the WMBB and the backbone network. Here, clients communicate with each other using multi-hop routing via intermediate clients and thus client WMNs are a type of MANET. Hybrid WMNs are a combination of infrastructure and client WMNs and constitute the most likely scenario. Here, mobile clients that are out of communication range of access points, may access the network through other clients that are in range.

2.2 Security Definitions

In this section, we briefly review some security concepts that are used in this thesis. For a general introduction to symmetric and public key cryptography as well as for a more detailed discussion of security properties, cryptographic primitives and protocols, please refer to [89]. Please refer to Table 2.3 for notations.

2.2.1 Some Cryptographic Primitives

Def. *Long-term and Short-term Credentials*: Long-term credentials contain authentic information to identify an entity and/or key material and are used over a longer period of time. On the other hand, short-term credentials, also called

Notations	
ID_i	Identifier of party i
N_i	Nonce chosen by party i
K_{ij}	Secret key shared between parties i and j
SK	Session key
$E_{K_{ij}}()$	Symmetric encryption under secret key K_{ij}
(Q_i, d_i)	Long-term public and private key pair of party i
(T_i, r_i)	Ephemeral public and private key pair of party i
$cert_i$	Public key certificate of party i's public key
$E_{Q_i}\{\}$	Public key encryption under i's public key
$S_{d_i}()$	Digital signature under i's private key
$f_{K_{ij}}()$	Keyed KDF
$f()$	Un-keyed KDF
$h()$	One-way hash function
$h_{K_{ij}}()$	MAC function
SID	Session identifier

Table 2.3: List of Notations: Authentication and Key Exchange Protocols

ephemeral credentials, may contain the same information but are changed frequently and used for a shorter period of time, e.g. one session.

Long-term credentials may be used in authentication protocols to prove an entity's identity or in key exchange protocols to derive session keys. Long-term credentials can be pre-shared secret keys or certified private and public key pairs, whereas short-term credentials are typically session keys or ephemeral private and public key pairs generated for one session. Short-term credentials are typically used to limit the damage of key compromise and reduce the amount of available ciphertext in a cryptanalysis.

Def. *Hash Functions*: A hash function $h()$ maps an arbitrarily long string to a string of fixed length ν, i.e. for binary strings $h() = \{0,1\}^* \mapsto \{0,1\}^\nu$. Secure hash functions satisfy two properties [38], they are *one-way*, i.e. it is computationally infeasible to find any input that maps to a pre-defined output, and *collision resistant*, i.e. it is computationally infeasible to find any two distinct inputs that map to the same output.

Def. ***Message Authentication Code (MAC)*****:** A MAC function maps, on input of a key K, an arbitrarily long string to a string of fixed length μ, i.e. $h_K() = \{0,1\}^* \mapsto \{0,1\}^\mu$. A MAC can be constructed from hash functions as keyed hash functions which is referred to as HMAC [39]. Computing an HMAC over an arbitrary input string ϕ, takes both key K and ϕ as input to a hash function $h()$, which we denote as $h_K(\phi)$, where K serves as random source to generate a random output with sufficient entropy.

Def. ***Key Derivation Function (KDF)*****:** A KDF function maps an arbitrarily long string to a string of fixed size ω, i.e. $f() = \{0,1\}^* \mapsto \{0,1\}^\omega$. KDFs can be constructed from hash functions and are keyed $f_K()$ or un-keyed $f()$ [1]. Keyed KDFs take an arbitrary input string ϕ and a key K as input, which we denote as $f_K(\phi)$, where K serves as random source to generate a random output with sufficient entropy.

Note that we distinguish MACs and keyed KDFs even though both are computed over a key and an arbitrary input string and can be constructed from hash functions. This is done to emphasize the purpose of the respective function in a security protocol, e.g. MACs can be used for providing authentication and integrity protection, whereas KDFs are used to derive cryptographic key material.

2.2.2 Identity-Based Cryptography

In 1984, Shamir proposed using user identities (IDs) or more precisely arbitrary strings as public keys [112] and introduced the first ID-based signature scheme. Using identities of users as their public keys has many implications. For example, user identities and their corresponding public keys do not need to be bound by certificates or any other means. Hence, public keys in ID-based cryptographic (IBC) schemes are self-authenticating. In IBC schemes the identity of a user i is represented as binary string $ID_i \in \{0,1\}^*$ of arbitrary length that contains information that unambiguously identify i, e.g. i's name, email address, date of birth, etc.. Identity ID_i can be used to derive i's public key Q_i with $Q_i = g_1(ID_i)$, where $g_1()$ is a publicly known function specified by each IBC scheme. Note that all users are able to derive the public keys of any other user in the network from

publicly known information without the need to exchange any data. This unique feature of IBC schemes is based on the assumption that the identities ID_i of all parties i in the network as well as function $g_1()$ are pre-known to all other parties in the network.

Due to the predetermination of public keys in IBC schemes, the private keys of users need to be generated and distributed by a TTP. Otherwise, users would derive their private keys from their public keys which would enable all users to compute the private keys of any other user in the network [112]. For that reason, IBC schemes require a TTP that serves as a key generation center (KGC) to generate and distribute private keys. Therefore, the KGC computes the private key d_i of each user i in the network using a master secret key s and the user's public key Q_i as inputs, i.e. $d_i = g_2(Q_i, s)$, where $g_2()$ is a publicly known function specified in each IBC scheme. Note that s is only known to the KGC and kept secret. The KGC then delivers the private keys to all users over a secure channel. We can observe that the KGC is a key escrow in IBC schemes due to its knowledge of all private keys in the network. We will discuss key escrow in great detail in Chapter 6.

2.2.3 Authentication and Key Exchange

In this section we introduce terminology and concepts of authentication and key exchange protocols. For a more detailed discussion refer to [18, 89].

Def. *Entity Authentication*: The process or protocol in which a party i provides evidence of its identity to another party j and j is assured that i actually participated in the process.

Upon executing an entity authentication protocol, j is assured that he is currently talking to i. We refer to the process as authentication for short. Throughout this thesis we consider protocols that provide *mutual authentication*, i.e. i and j mutually authenticate each other. Note that two unilateral authentication protocols, each executed in one communication direction, are not sufficient for providing mutual authentication because it is not clear whether both parties participated in both protocols. To resolve this problem, the authentications in both directions need to be interleaved. A common way to provide (mutual) authentication is the

challenge-response technique [89] in which a party j successfully authenticates himself to a party i by sending a correct response to i's challenge. To prevent replay attacks, challenges and responses contain either a nonce, i.e. a random number that is used once, or a time stamp. Note that due to the lack of a central clock and the difficulties of synchronizing clocks in MANET, we consider nonces as the only suitable choice in MANETs. Challenge-response based mutual authentication protocols using nonces require at least three message flows and can be implemented using symmetric or public key cryptography [89].

In a symmetric challenge-response protocol, both parties authenticate each other by providing evidence of their knowledge of a pre-shared key K without revealing the key. For example, parties could encrypt a challenge or decrypt a challenge that was encrypted under K. In another approach using symmetric primitives, parties compute a MAC over a challenge.

There are two general approaches for providing challenge-response based mutual authentication using public key techniques. In the first one, parties i and j prove the possession of their private keys by decrypting a challenge encrypted under their respective public keys. In the other method, parties i and j each sign a challenge using their private keys.

Def. *Key Exchange*: The process or protocol whereby a shared key becomes available to two parties for subsequent cryptographic use.

If two or more parties wish to protect their communications they need to establish a fresh session key, e.g. using their long-term credentials in some key exchange protocol. The established key material, commonly referred to as session key, can be used to derive further keys that can be used to encrypt, authenticate and/or integrity-protect all further communications during the current session. Key exchange protocols can be based on symmetric or public key cryptography and many protocols exist [18, 89].

Protocol 1. EC-DH Key Exchange Protocol

PROTOCOL FLOW:

1. $i \longrightarrow j : T_i = r_i P$
2. $i \longleftarrow j : T_j = r_j P$

SESSION KEY: $SK = r_i T_j = r_j T_i$

Diffie-Hellman (DH) Key Exchange: We now briefly discuss the Diffie-Hellman (DH) key exchange protocol [34] as example of a public-key based key exchange protocol. More precisely, we review an elliptic curve variant of the protocol (referred to as EC-DH key exchange [96]) in which ephemeral public keys are exchanged to derive a fresh session key SK. This protocol serves as building block in many protocols proposed in the remainder of this thesis and the protocol flow is illustrated in Protocol 1. We introduce the following notation, let $E(\mathbb{F}_q)$ be an elliptic curve over a finite field $\mathbb{F}_q$, where q is a prime or multiple of a prime and P a generator of $E(\mathbb{F}_q)$, where all parameters are chosen according to [96]. Now two parties i and j executing an EC-DH key exchange, each compute an ephemeral public key with $T_i = r_i P$ and $T_j = r_j P$, respectively, where $r_i, r_j \in \mathbb{F}_q$ are randomly chosen in each protocol execution and serve as ephemeral private keys. Then both parties exchange their ephemeral public keys and compute the DH session key as $SK = r_i T_j = r_j T_i$.

Protocol 2. MAC-based AKE Protocol

PROTOCOL FLOW:

1. $i \longrightarrow j : ID_i, SID, X_i$
2. $i \longleftarrow j : j, SID, N_j, h_{K_a}(ID_i, N_i, SID, N_j)$
3. $i \longrightarrow j : ID_i, SID, r_i = h_{K_a}(ID_j, N_j, SID, N_i)$

Def. *Authenticated Key Exchange*: Key exchange protocol providing mutual authentication to ensure that established keys are authentic, i.e. both parties know

who they share the established session keys with. These kind of protocols are referred to as *authenticated key exchange (AKE) protocols*.

Protocol 3. Signature-based AKE Protocol

PROTOCOL FLOW:

1. $i \longrightarrow j : ID_i, SID, X_i$
2. $i \longleftarrow j : ID_j, SID, X_j, S_{d_j}(ID_j, SID, X_j, X_i, ID_i)$
3. $i \longrightarrow j : ID_i, SID, S_{d_i}(ID_i, SID, X_i, X_j, ID_j)$

As authentication protocols, AKE protocols can be implemented using symmetric or public key cryptographic primitives. As proposed in [25], we distinguish three general approaches for providing authenticated key exchange: MAC-based (see Protocol 2), digital signature-based (see Protocol 3), and public key encryption-based (see Protocol 4) AKE protocols. In addition, the session key can be derived in a symmetric or public key-based manner. First approach uses a keyed KDF with shared key derivation key K_d and fresh mutually exchanged nonces as input, i.e. $SK = f_{k_d}(N_i, N_j)$, whereas the second approach may use a DH-based key exchange with $SK = r_i T_j = r_j T_i$ in EC-DH. Note that K_d may be derived from a secret key K_{ij} that is shared between two parties i and j. For example, $K_d = f_{K_{ij}}(2)$ in Protocol 2 and $k_d = K_{ij}$ in Protocols 3 and 4. Authentication key K_a required in MAC-based AKE protocols can be derived as $K_a = f_{K_{ij}}(1)$ for symmetric key derivation and $K_a = K_{ij}$ for public key derivation. In the illustrated AKE protocols 2-4, $X_i = N_i$ and $X_j = N_j$ for symmetric key derivation and $X_i = T_i$ and $X_j = T_j$ for public key derivation using EC-DH.

Protocol 4. Public Key Encryption-based AKE Protocol

PROTOCOL FLOW:

1. $i \longrightarrow j : ID_i, SID, E_{Q_j}\{X_i\}$
2. $i \longleftarrow j : ID_j, SID, E_{Q_i}\{X_j\}, h_{X_i}(ID_i, SID, E_{Q_i}\{X_j\})$
3. $i \longrightarrow j : ID_i, SID, h_{X_j}(ID_j, SID, E_{Q_j}\{X_i\})$

We now list some mandatory and desirable security properties of AKE protocols, where we adopt the definitions from [19, 27]. The protocol properties prevent most common attacks on AKE protocols. All AKE protocols executed by two parties i and j should achieve at least the three following necessary properties (NP):

- *(NP-1) Mutual entity authentication:* Ensures that i and j mutually authenticate each other.
- *(NP-2) Mutual implicit key authentication:* Ensures that the established key is only known to i and j, i.e. the session key is kept confidential. Key authentication also implies that the key is fresh, since a key that is not fresh cannot be guaranteed to be confidential.
- *(NP-3) Completeness:* Ensures that i and j are both deriving the same session key after a successful protocol execution.

In addition to the necessary security properties, we consider the following desirable security properties (DP):

- *(DP-1) Known-key security:* Ensures that even if one or more expired session keys are compromised the adversary cannot compute new, i.e. currently used, session keys or gain any other secret information such as long-term private keys. In other words, this property prevents known-session key attacks.
- *(DP-2) Unknown key-share (UKS) resilience:* Prevents so-called unknown key-share or identity-misbinding attacks. An adversary O cannot fool a principal j to think that he shares the key with O, although j actually established

the key with i. Hence, after the protocol execution two principals i and j are ensured that they both share a key with each other.

- *(DP-3) Key control:* Indicates that all communication parties contribute to the session key computation. In that way all principals can ensure that the generated session key is fresh and has good random properties.
- *(DP-4) Deniability:* Opposite of non-repudiation which is achieved if a principal i, upon executing a protocol run with a party j, cannot prove to a third party that she has indeed communicated with j.
- *(DP-5) Key compromise impersonation (KCI) resilience:* During a KCI attack an adversary first compromises a long-term private key of a party, say i, and then masquerades as a different principal, say j, to i. The property of KCI resilience prevents adversaries from impersonating other network principals than the compromised ones.
- *(DP-6) Perfect forward secrecy (PFS):* Achieved if long-term private keys of some principals are compromised and expired session keys that have been previously established between the same principals are not compromised too. This feature cannot be achieved by protocols based on purely symmetric primitives and is usually achieved by executing a DH-like key agreement.
- *(DP-7) TTP-PFS:* Necessary stronger notion of PFS in ID-based schemes that considers the compromise of the system's master key s that is only known to the TTP. If TTP-FS is provided, an adversary cannot obtain any expired session keys of previous sessions even if he is in possession of s.
- *(DP-8) Non-Repudiation:* Opposite of deniability, which ensures that parties involved in a protocol execution cannot deny sending message they have committed to. Non-repudiation implicitly provides integrity protection and message authentication.
- *(DP-9) Replay resilience:* An adversary cannot replay messages from previous sessions to impersonate a user.

Note that DP-9 is already covered by NP-1 and NP-2 and only listed for completeness.

Def. *Pre-Authentication*: Initial exchange of credentials to establish pre-shared long-term credentials.

Pre-authentication is a necessary prerequisite for authentication and/or key exchange protocols. For instance, in symmetric AKE protocols, both parties need to pre-share a secret key, whereas in public key-based AKE protocols, both parties need an authentic copy of each others public key. The same long-term credentials are used in all authentications and/or key exchanges between same pairs of nodes. The difficulty of providing pre-authentication is based on the problem of establishing a protected channel for secure credential exchange without sharing any credentials. Note that "protected" refers to authentic for exchanging public keys and authentic and confidential for exchanging secret keys. We will discuss methods of providing protected channels for pre-authentication in MANETs in Chapter 3. Pre-authentication in MANETs can either occur during network or node initialization, i.e. the TTP helps to establish the pre-shared credentials among pairs of nodes, or in the running network, i.e. pairs of nodes need to establish a protected channel without any external help.

Def. *Key Distribution*: Distribution of individual and pre-shared keys by a TTP as part of node and/or network initialization.

Def. *Key Revocation*: Mechanism to revoke expired or compromised keys and making this information available to all network nodes.

Revocation information is typically provided in form of lists, e.g. a blacklist containing all revoked certificates or a so-called whitelists with all valid certificates. Sometimes both lists are combined. Typically the lists are generated by a CA and then either pushed to all nodes or made publicly available to nodes (pull approach). In some cases nodes can request the status of specific certificates (pull). A widely deployed revocation scheme for X.509 certificates uses certificate revocation lists (CRLs) [103], which are blacklists that are generated by a CA and stored in publicly accessible repositories. Network nodes can then access the repositories to download the latest CRL. Many schemes were introduced to reduce the size of the CRLs that

can grow very large over time. In the simplest approach, only updated information is provided in the next published CRL, in so called *delta CRLs*. Another popular revocation scheme is the Online Certificate Status Protocols (OCSP) [104], in which network nodes request the status of particular certificates and a CA returns the signed status of the requested certificates.

We can observe that these solutions require a fixed infrastructure, such as a CA and/or public repositories, to generate and distribute the revocation information. Consequently, network nodes that want verify whether a certificate is revoked must have access to this infrastructure. Hence, existing revocation solutions that are widely used for infrastructure networks such as LANs and WLANs are not suitable for a deployment in MANETs. In fact, providing certificate or key revocations in MANETs is one of the most challenging problems in all MANETs that employ public key schemes. We will address the revocation problem in Chapter 5.

Def. *Key Renewal*: Process of obtaining a new key upon the previous one has expired or been revoked.

Hence, key renewal algorithms are necessary to complement revocation schemes. In traditional infrastructure-based networks, nodes may simple re-authenticate to a CA or other TTP and obtain a new key. However, providing key renewal is not as straightforward in MANETs and solutions rely on the availability of a TTP. We present a key renewal scheme for MANETs in Chapter 5.

Def. *Key Escrow*: An entity that is in possession of some or all secret or private keys in the network.

Key escrows are able to decrypt communications and may be able to impersonate nodes. For example the KGC in IBC schemes is a key escrow. This feature is typically considered as a drawback but sometimes looked at as a desirable feature. We analyze key escrow in MANETs in Chapter 6

Chapter 3

Pre-Authentication Models

As mentioned in the previous chapter, pre-authentication is a crucial prerequisite for all networks intended to support authentication and key exchange among network nodes. In this chapter, we categorize and analyze several pre-authentication models (PAMs) for MANETs. Basically, a PAM describes a method of how a secure channel for pre-authentication can be established. For each presented model, we identify the deployment conditions, reference existing protocols that have been introduced in this model and discuss the model's applicability and limitations. For easier comparison, we provide a summary of all presented pre-authentication models in Table 3.1. In Section 3.3 we summarize parameters of our target MANET applications in this thesis and derive design goals for security solutions for such applications. These design goals help to identify a suitable pre-authentication model in the last section, where we discuss the applicability of all discussed pre-authentication models. Please note that parts of our discussions are taken from our previous publications in [52, 58].

3.1 Secret Key-Based Solutions

When employing security protocols based on symmetric cryptography, a secret must be shared among each pair of network nodes. We distinguish between secret key solutions in which all nodes share the same key and solutions in which pairs of

Model	Implementation	Comments*
PAM-S1. Black Box	Keys exchanged over secure side-channel, e.g. EAP-GPSK [30]	− secure channel not specified
PAM-S2. Administrator	Key manually entered in all nodes, e.g. Bluetooth [15], ZigBee [71], WEP [68]	− does not scale well − single node can compromise network
PAM-S3. Pairwise Pre-Distribution	nodes initialized with (subset of) keys before deployment, e.g. [36]	+ limits damage by compromises + scalable − new nodes cannot join
PAM-S4. Physical Contact	Key exchanged by physical contact, e.g. [115]	+ simple + no infrastructure − requires proximity of nodes
PAM-P1. Location-Limited	Public keys directly exchanged, e.g. [7, 23]	+ no certificates + no infrastructure − requires proximity of nodes −pre-existing trust
PAM-P2. ID-Based	Identity used as self-authenticated public key, e.g. [33, 53, 77, 124]	+ no certificates + no message exchange + implicit pre-authentication − key escrow
PAM-P3. Self-Certified Public Key	Certificate embedded in public key	+ no certificates + implicit pre-authentication − no proposed solutions
PAM-P4. PKI	PKI in MANET, e.g. using (k, n)-threshold scheme [78, 85, 125]	+ simple pre-authentication + self-organized† − exchange of certificates − set up not efficient‡
PAM-P5. Trusted Path	Every node is own TTP; PGP-like, e.g. [26, 67]	+ self-organized − not efficient − pre-existing trust

Table 3.1: Pre-Authentication Models for MANETs

*"+"/"−" denote advantages and disadvantages of the model, respectively.

†In case of (k, n)-threshold internal distributed CA.

‡Efficiency with respect to computation and communication costs.

nodes share a key. These long-term secrets can be established during the network initialization phase, before a node joins an existing network, or in the running network. In first two cases, pre-authentication is done between the TTP and a node, whereas in latter case pre-authentication is performed by pairs of nodes. Either way, an authentic and confidential channel needs to be established to ensure secure pre-authentication. The following models for secret key-based pre-authentication (PAM-S) describe how such a secure channel can be established.

PAM-S1. Black Box Model

In this model, it is assumed that the secret keys are exchanged over a secure side-channel during the pre-authentication phase. However, it is not further specified how this secure channel is achieved. This black box model is assumed in many authentication, key exchange and other security protocols for MANETs. These solutions simply assume the existence of pre-shared secrets and leave it up to the administrator or users how to accomplish pre-authentication. For example, the security amendments in IEEE 802.11i [69] allow the use of secret key-based EAP methods [107], such as EAP-GPSK [30], which do not specify how these keys are shared.

PAM-S2. Administrator Model

In this model, every new node is set up with the same network key prior joining the network. The network key can be a password, PIN, or a cryptographically strong key. Typically the key is manually entered in all nodes by an administrator or a user of the network. It must be ensured that used keys have sufficient entropy to prevent dictionary attacks. Otherwise, stronger keys need to be securely derived from initially shorter and more user friendly passwords, e.g. [4,11]. Here the secure channel consist of a trusted person acting as a TTP and physically entering a key in authorized nodes. This solution is applicable to small networks, such as WPANs. These networks can be managed by a single administrator, e.g. one user connecting several of his devices to form a network, such as laptop, PDA, keyboard, printer, headset, etc. This model is not applicable to larger networks because it does not scale well. Another limitation of this approach is the fact that the compromise of

a single key leads to the compromise of the entire network. This poses a serious concern in MANETs due to the likelihood of node compromises (see Section 2.1.2).

This model has been applied to several standards for wireless networks. For instance, in IEEE 802.15.1 for WPANs, also known as Bluetooth, users can set up a so-called piconet consisting of up to 8 devices by entering the same key in all devices. The same idea applies to low rate WPANs (LR-WPANs) specified in IEEE 802.15.4, also referred to as ZigBee, where an administrator sets up all devices with the same key. Another examples is IEEE 802.11 for WLANs where users manually enter a WEP key in all devices.

PAM-S3. Pairwise Pre-Distribution Model

As in the previous model PAM-S2, a TTP distributes the secret keys to nodes before they join the network. However, to limit the overall damage of compromised nodes, this model uses pairwise shared keys instead of a single network key. Key initialization is done for all nodes at once in an automatic process, e.g. all nodes are programmed with the keys before released into the network. In a variant, each node is initialized with a subset of all pairwise keys which helps to save scarce memory space for sacrificing that each pair of users pre-shares a key. In that case, pre-sharing becomes probabilistic and protocols are necessary to ensure that nodes that do not pre-share keys can establish keys [36, 84]. The efficiency of these type of solutions suits the resource constraints of MANET devices. For this reason this approach is attractive for WSNs which are typically severely resource constrained. However, the suitability of this approach for MANET applications is rather limited. For instance, the model requires that all nodes are present at the network initialization. While this is common in WSNs (where nodes are typically deployed at the same time), MANET applications are generally more dynamic and nodes may join or leave at any time. Pre-distribution does not provide the necessary flexibility and scalability to allow nodes to join the network after initialization and securely communicate with any network node. Furthermore, pairwise pre-distribution requires a trust model that is different from public key-based models. For instance, former approach requires a central external TTPs for key initialization, where the TTP knows all secret keys in the network. On the other hand, latter approach can have external

or internal TTPs, where the TTP is trusted to verify and certify the authenticity of nodes but does not know the private keys.

PAM-S4. Ad Hoc Model
In this model, pre-authentication is provided in the running network, i.e. nodes are not initialized by a TTP during network initialization or prior joining the network. Instead, pairs of nodes exchange credentials whenever they wish to communicate. This implies that an authentic and confidential channel must be established within the network without the help of any shared credentials. Without the help of additional credentials, there is no other choice but to exchange the secret keys in plaintext. However, to prevent attacks by eavesdroppers the two nodes need to be in close proximity to each other. If the key is transmitted over an one-hop connection and nobody else is in communication range of these nodes, the channel provides confidentiality and authenticity. The best way to achieve this channel is by physical contact. The idea of exchanging secrets by physical contact was first proposed in [115].

We can observe that this is the only true ad hoc pre-authentication model for secret keys, because it does not require any TTP or any other infrastructure. Since pair-wise keys are used as opposed to network keys, node compromise is limited to keys shared by the compromised node and does not compromise the entire network. In addition, the method is very efficient. However, the applicability of this solution is very restrictive and only suitable for applications that provide close proximity of communicating nodes.

3.2 Public Key-Based Solutions

In this section, we describe several public key-based pre-authentication models (PAM-P) that provide methods to obtain an authentic copy of another node's public key. Due to better, more flexible and scalable key management and thus wider applicability, most research focuses on public key based security for MANETs. In addition, only public key-based solutions enable the use of digital signatures, which are the only cryptographic method to provide non-repudiation in communications.

We summarize some work on public-key based solutions for MANETs as part of our discussion. Unlike pre-authentication channel for PAM-S, authentic channels are sufficient because public keys do not require confidential transmission.

PAM-P1. Location-Limited Model

If close proximity of network nodes is given, a secure pre-authentication channel can be established by visual or physical contact among communicating nodes. Such pre-authentication channels enable nodes to directly exchange their public keys, i.e. without the necessity of a TTP and public key certificates. Please note that eavesdroppers do not pose a threat when exchanging public keys and thus the conditions for such location-limited channels are less stringent than in PAM-S4. However, a node needs to be certain that the received public key was received from the claimed node and not from somebody else. This model is based on two assumptions: (1) nodes that wish to communicate are in close proximity to each other; and (2) all nodes know which other nodes are trustworthy. Latter assumption is based on the fact that no TTP vows for the credibility of the other node, e.g. by issuing a certificate. Consequently, nodes need to recognize each other and some pre-existing trust must exist among nodes. The model works well in all applications that meet these two assumptions and is not feasible in others. Protocols in this model have been introduced in [7, 23]. Note that if nodes are able to perform physical contact, exchanging secret keys is preferable (see PAM-S4).

PAM-P2. Identity-Based Model

IBC schemes use identities as public keys which makes public key certificates redundant. A KGC is required to generate and distribute private keys to all nodes. This distribution occurs during network initialization and node initializations. Since nodes do not need to exchange public keys as long as they know the identities of each other, there is no need for pre-authentication channels among network nodes. This feature makes the ID-based model attractive for MANETs. We would like to point out that depending on the application and what kind of information is used as "identity", it may or may not be reasonable to assume that all nodes know each others identities ahead of time. If identities are not pre-known in the network, they

must be exchanged prior communication. However, identities can be exchanged over a completely unprotected pre-authentication channel, as long as nodes trust the KGC that issues the corresponding private keys. For instance, a user may trust all identities that are email addresses from a trusted domain, such as a company or university. Or a node trusts a MAC or IP address from a group of trusted addresses. For a discussion of suitable identity strings please refer to Section 4.3.1.

On the other hand, if identities are pre-known, pre-authentication implicitly takes place during network initialization between KGC and each node. Note that this channel needs to be confidential and authentic because private keys are distributed, as opposed to public keys as in other PKI schemes. The channel conditions can be relaxed to authentic channels by using a blinding technique as shown in [81]. A well known problem of all ID-based scheme is key escrow, because the KGC is in possession of the private keys of all network nodes. Some ID-based protocols and solutions have been recently introduced for MANETs, e.g. [33, 53, 77, 124].

PAM-P3. Self-Certified Public Key Model

Self-certified public keys in which the certificates are embedded in the public keys themselves have been introduced in [43]. Consequently, only public keys but no certificates need to be exchanged among nodes. In fact, a node's identity is part of its public key and signed by a TTP and the node itself. A TTP is required to generate and distribute the self-certified public keys either during network or node initialization. The pre-authentication, i.e the exchange of public keys occurs in the running network. Because the authenticity of the public keys is provided by the keys themselves, pre-authentication does not require a secure channel. No protocols or solutions for MANETs have been introduced in this model.

PAM-P4. PKI Model

In this model a conventional PKI scheme is deployed. Nodes exchange public key certificates during pre-authentication, where the certificates help to establish an authentic channel. If an external CA is implemented, public key certificates are distributed to all nodes prior joining the network. The nodes then exchange these certificates with each other in the running network. Public key certificates can

be relatively long and thus create some communication and storage overhead that may conflict with node constraints. It is infeasible for nodes to check whether certificates have been revoked or to renew certificates. Note that the same is true for all other models employing external TTPs, e.g. PAM-P2 and PAM-P3. To enable these functionalities, a distributed internal CA needs to be deployed. Here, a CA is emulated by k out of n nodes using a (k, n)-threshold scheme [111]. A lot of the early research on MANET security has focused on distributed internal CAs, e.g. [78, 85, 125]. Note that the use of threshold schemes introduces a lot of additional computational and communication overhead. Furthermore, threshold schemes are quite cumbersome to implement and require a fairly large number of nodes to work well. The same problems occur in other schemes, e.g. ID-based schemes, when threshold schemes are employed to emulate an on-line TTP.

PAM-P5. Trusted Path Model

In the trusted path model, network nodes issue and distribute their own certificates and sign other certificates in a PGP manner [102]. In that way no external or internal TTPs are necessary. Pre-authentication is done within the running network by exchanging public keys. To ensure that public keys are authentic, a node needs to find a chain of signed certificates from the node that sent the key back to the node that received the key. This model emphasizes the self-organization property of MANETs and assumes the existence of trust among some nodes. The performance of pre-authentication highly depends on the length of the trusted path, which is generally hard to predict. This approach is very efficient in the set-up phase and, unlike schemes using threshold schemes, does not require computations by any nodes but the communicating ones. However, a node probably needs to verify more than one certificate for pre-authentication. An example of a proposed protocol in this model is [67]. This model is also applied to a group case, in which trusted subgroups search for intersections to create a trusted path [44].

3.3 Parameter Choices and Design Goals

As mentioned earlier, the properties of MANETs are universal for all applications and dictate some general design goals for MANET protocols. Other design goals are determined by the parameter choices of specific applications. In this section, we first specify the parameter choices for our security solutions that we will present in the following chapters. Next, we summarize the design goals for security protocols that are designed to meet the special properties of MANETs as well as the selected parameters. The derived design goals help to identify suitable pre-authentication models as well as cryptographic schemes for authentication and key exchange. All presented security solutions in this thesis will be designed according to the derived design goals.

We assume the following network parameters for the targeted MANET applications:

- flat topology, i.e. there is only one layer of network nodes and all nodes have similar capabilities
- non-controlled, i.e. there are no controller nodes in the network and all nodes have identical roles
- one domain, i.e. we assume all nodes are from the same domain and there is only one (central or distributed) TTP
- nodes have moderate resource constraints, e.g. nodes are capable of executing a few demanding cryptographic operations, such as bilinear pairing computations and other public key operations. Furthermore, nodes have non-volatile memory large enough to store system parameters and cryptographic keys.
- no location awareness, i.e. nodes neither know their location nor can prove their location.
- TTP availability AV-2, i.e. an external TTP is available to initialize all network nodes. However, we show our presented solutions can be extended to cases AV-1, AV-3 and AV-4.

- no pre-existing trust, i.e. there are no existing relationships among nodes and nodes only need to trust the TTP.
- mid-sized networks with average sizes between two to 100 nodes. This excludes simple solutions for consistently small networks and requires some degree of scalability.
- no protected out-of-band channels for pre-authentication, i.e. solutions should not require protected pre-authentication channels in the running network.

From the MANET properties as listed in Section 2.1.3 and the selected network parameters above, we derive the following design goals for our authentication and key exchange framework:

1. few computational steps per node
2. mostly lightweight computation steps; only a few more demanding operations
3. few number of message exchanges
4. small messages
5. small program and data storage requirements
6. limited consequences of data disclosure
7. balanced protocols, i.e. participants perform similarly hard and many operations
8. support node mobility
9. support of subsequently joining nodes
10. scalability
11. self-organized solutions for pre-authentication, authentication, key exchange and key revocation, only initialization and key renewal require access to TTP

12. no pre-existing trust among nodes assumed

13. pre-authentication does not require protected channels among nodes

The first six design goals are addressing the resource constraints of MANET devices as summarized in Section 2.1.2. The seventh design goal addresses the fact that there is neither a controller nor layers with more powerful nodes in the network. Goals number 8 and 9 address the mobility and dynamic of MANETs, respectively. The remaining goals describe design goals specific to the targeted MANET applications, namely middle-sized MANETs consisting of moderately powerful devices, such as PDAs, pockets PCs or laptops, in which a TTP initializes all nodes before joining the network and all network operations are executed without the help of any infrastructure. Only key renewal requires accessing a TTP. No trust or trusted channels exist among the devices.

3.4 Discussions and Conclusions

We now summarize the applicabilities and limitations of the presented pre-authentication models. We then point out which of the models is best suited to provide pre-authentication in the targeted MANET applications (see Section 3.3).

Due to their superior efficiency, symmetric schemes seem more suitable for MANETs. However, the difficult key distribution significantly reduces the applicability of symmetric schemes. As mentioned earlier, the black box model (PAM-S1) does not provide a secure pre-authentication channel and emphasizes the importance that every symmetric security protocol should provide or at least discuss how pre-shared keys are exchanged. We conclude that symmetric schemes only work for very small networks with a single administrator (PAM-S2), networks in which all nodes are in close proximity to each other (PAM-S3) or for WSNs in which a TTP initializes all nodes before their deployment (PAM-S4).

Public key based schemes are computationally less efficient but provide more flexibility and ensure that node compromise does not affect uncompromised network nodes. We distinguish four categories: (1) with TTP and public key certificates, e.g.

PAM-P4; (2) with TTP and no public key certificates, e.g. PAM-2 and PAM-3; (3) without TTP but with public key certificates, e.g. PAM-P5; and (4) without TTP and no public key certificates, e.g. PAM-1. If proximity of nodes is provided, PKI schemes without certificates can be used in MANETs. If proximity is not provided, nodes need to exchange public key certificates. Avoiding the use of certificates without the requirement of node proximity can be achieved by using self-certified or ID-based public keys, respectively. First approach still requires the exchange of the keys, whereas latter does not require any key exchange at all.

In this thesis, we consider midsized MANETs, i.e. networks that are larger than typical PANs yet smaller than WSNs, where we neither assume pre-existing trust among nodes nor close proximity. This leaves us with PAM-P2, PAM-P3 and PAM-P4 as potential solutions. Among these, the ID-based model seems most attractive due to its implicit pre-authentication that does not require the exchange of any messages among nodes. For this reason we choose the ID-based model as basis for developing a security framework for the targeted MANET applications.

We would like to note that in all models, nodes cannot verify whether a key (secret or public) has been revoked. The revocation problem has been addressed in the PKI model by introducing a distributed internal CA. This approach can be applied to other models as well. However, as pointed out earlier, threshold schemes introduce a large overhead to the network which is undesirable in MANETs. For this reason we introduce a self-organized revocation scheme in this thesis that does not require any internal TTP.

Chapter 4

Certificateless Authentication and Key Exchange Framework for MANETs

Authentication and key exchange are both essential security objectives in any computer network, including MANETs. In this chapter, we propose a certificateless authentication and key exchange framework for MANETs. Our framework provides pre-authentication among nodes and enables the use of certificateless authentication and key exchange protocols. The framework can be used with any ID-based authentication, key exchange or other security protocol from bilinear pairings. The solution is especially designed to suit all MANETs properties and design goals as specified in Sections 2.1.3 and 3.3, respectively. We choose IBC schemes for developing our security framework because of their special properties enabling efficient pre-authentication in MANETs as discussed in the previous chapter. Unlike conventional PKI schemes, IBC schemes do not create any storage and communication overhead for storing and exchanging public key certificates, respectively. IBC schemes have some other features that are beneficial for a deployment in MANETs which we discuss later in this chapter. In the remainder of this chapter, we first review bilinear pairing-based IBC schemes and discuss special features and challenges of such schemes when deployed in MANETs. Furthermore, we review some

previous work, before introducing our ID-based authentication and key exchange framework in Section 4.3. In Section 4.4, we introduce a set of ID-based AKE protocols. In Sections 4.5 and 4.6, we analyze the security and performance of the proposed framework and protocols. Finally, in Section 4.7 we draw conclusions. Earlier versions of our authentication and key exchange framework appeared in [57, 60] and the ID-based AKE protocols can be found in [53].

4.1 Review Bilinear Pairing-Based IBC Schemes

In 2001 Boneh and Franklin introduced the first ID-based encryption (IBE) scheme from the Weil pairing [17] which we refer to as *BF-scheme* in the remainder of this thesis. Much research on ID-based schemes from Weil and other bilinear pairings has been carried out ever since. Proposed protocols from bilinear pairings include encryption, signature and authentication schemes, e.g. [17,51,87], respectively. The ID-based authentication and key exchange framework including the ID-based AKE protocols proposed later in this chapter are all based on the BF scheme. We now review bilinear pairings and the BF scheme, where we adopt most notations from [17].

4.1.1 Bilinear Pairings

In the following, we give a definition of cryptographic bilinear mappings, the building block of the BF scheme and all other pairing-based IBC schemes.

Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two groups of the same prime order q. $\mathbb{G}_1$ is as an additive group, i.e. a group of points on an elliptic curve, whereas $\mathbb{G}_2$ is a multiplicative subgroup of a finite field. Let P be an arbitrary generator of $\mathbb{G}_1$. Assume that discrete logarithm problem (DLP) is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$. A bilinear mapping $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ must satisfy the following properties:

- **Bilinearity:** $\hat{e}(aP, bQ) = \hat{e}(P, Q)ab$ for all P, $Q \in \mathbb{G}_1$ and a, $b \in \mathbb{Z}_q^*$.
- **Non-degeneracy:** If P is a generator of $\mathbb{G}_1$, then $\hat{e}(P, P)$ is a generator of $\mathbb{G}_2$. In other words, $\hat{e}(P, P) \neq 1$.

- **Computable:** There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for all P, $Q \in \mathbb{G}_1$.

Weil pairings [17] and Tate pairings [8] are examples of such cryptographic bilinear mappings, where Tate pairings are computationally more efficient and thus preferable in MANETs.

4.1.2 Parameter Generation and System Set Up

The security of pairing-based IBC schemes is based on the so-called Bilinear Diffie-Hellman Problem (BDHP) [17] which is defined as follows.
***Bilinear Diffie-Hellman Problem (BDHP)*:** Given two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ and a generator P of $\mathbb{G}_1$, the BDHP is to compute $\hat{e}(P, P)^{abc} \in \mathbb{G}_2$ for any $a, b, c \in \mathbb{Z}_q^*$ given (P, aP, bP, cP).

In any IBC scheme, the KGC is responsible to select the system parameters and set up the system such that the BDH problem is hard. In the following paragraphs, we review the set up and extract algorithms of the original BF scheme, which describe the secure set up of a pairing-based IBC scheme and secure derivation methods for ID-based keys. The same algorithms serve as a basis in our authentication and key exchange framework. However, we will not review the encryption and decryption algorithms, because they are note required in our framework.
Setup:

1. On input of a security parameter k, the KGC uses a BDH parameter as defined in [17] to generate a prime q, two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$. The KGC chooses a random generator $P \in \mathbb{G}_1$.

2. The KGC picks a random $s \in \mathbb{Z}_q^*$ and sets $P_{pub} = sP$.

3. The KGC chooses a hash function that maps an arbitrarily long binary string to an element in group $\mathbb{G}_1$, i.e. $H_1 : \{0, 1\}^* \mapsto \mathbb{G}_1^*$.

The KGC publishes $param =< \mathbb{G}_1, \mathbb{G}_2, q, P, \hat{e}, P_{pub}, H_1 >$ as public parameters and keeps s secret. The KGC's private and public key pair is (s, P_{pub}).
Extract: The public key $Q_i \in \mathbb{G}_1^*$ of a node i with identity ID_i is as

$$Q_i = H_1(ID_i), \tag{4.1}$$

where ID_i is an arbitrarily long binary string $\in \{0,1\}^*$. The KGC derives the private keys d_i for every node i as

$$d_i = sQ_i. \tag{4.2}$$

We can observe that instead of directly using identities as public keys, as done in Shamir's scheme, here the identity string is first mapped to a point on an elliptic curve using hash function H_1. Note that public keys can be computed from publicly available information, whereas private keys can only be computed by the KGC because the computation requires the KGC's private key s as input. The KGC generates the private keys and securely distributes them to the nodes. The key distribution channel between KGC and nodes needs to be authentic and confidential. However, if a blinding technique such as in [81] is used, an authentic channel is sufficient.

To limit the validity period of an ID-based public key, an expiry date can be embedded in the key itself. This can be done by concatenating an expiry date t_x to the public key [17], with

$$Q_i = H_1(ID_i||t_x) \tag{4.3}$$

for the public key Q_i of user i. Only if user i is in possession of the matching private key that corresponds to the correct date, i.e. $d_i = sH_1(ID_i||t_x)$, he can sign or decrypt messages. The granularity of the validity period is a system parameter that describes a security/efficiency trade-off. For instance, a shorter period reduces the risk of key compromise but induces more overhead because users need to frequently obtain fresh private keys from the KGC [86].

In addition to pre-shared public keys, each pair of users i and j in a pairing-based IBC scheme is able to compute a pairwise pre-shared secret key K_{ij} with

$$K_{ij} = \hat{e}(d_i, Q_j) = \hat{e}(Q_j, d_i) \quad (4.4)$$

$$= \hat{e}(Q_i, Q_j)^s \quad (4.5)$$

in a non-interactive fashion [108]. Such pre-shared secret keys have been used in ID-based authenticated encryption schemes [87] and AKE protocols [19]. For the key computation both parties compute the bilinear mapping $\hat{e}(\cdot)$ over their own private key d_i and the public key Q_j of the desired communication partner. Note that the KGC is able to compute all pre-shared keys according to Eq. (4.5).

4.2 IBC Schemes Employed in MANETs

After providing a general overview, we now discuss the deployment of IBC schemes in MANETs.

4.2.1 Distinctive Features and their Benefits to MANETs

We believe that IBC schemes are an attractive security solution for many MANET applications, including our targeted applications. IBC schemes provide the following special features:

1. *implicit* and *non-interactive* pre-authentication among all network nodes
2. *implicit* public key validity checks

The first feature is due to the use of identities as public keys which entails many desirable properties. IBC schemes do not require any secure channel for pre-authentication because public keys are self-authenticating and known prior to communication. In contrast to public key certificates in PKIs, no additional credentials to proof the authenticity of keys are needed. The communication overhead is reduced because public keys do not need to be exchanged.

The second feature provides an easy way to check whether a public key is valid. We denote a key as *valid* if the key is not expired. As shown in Eq. (4.3), the expiry date can be directly embedded in the public keys. When verifying a signature in an ID-based signature scheme, we check the validity of the keys at the same time, whereas in ID-based encryption schemes, only users with valid keys can decrypt. In contrast, in PKI schemes expiry dates are listed in public key certificates and thus nodes can still decrypt or sign even if their key is expired. Here, nodes need to explicitly check the expiry date in a certificate to see whether a key is expired or not.

All pairing-based IBC schemes offer an additional features that is attractive for MANETs:

3. every pair of nodes i and j pre-share a pairwise secret key K_{ij} (see Eq. (4.4)) in a *non-interactive* fashion

This additional feature of paring-based schemes offers all the benefits of symmetric key schemes without the need of a secure channel during pre-authentication. Each pair of users i and j in the network shares a secret K_{ij}, before ever having communicated with each other. The pre-shared secret keys can be used to enable the use of symmetric mutual authentication, key exchange and other security protocol at low computational and communications costs. Please note that ID-based pre-shared keys can only be computed in a non-interactive fashion if the identities are pre-known. Otherwise, identities must be first exchanged. In that case we loose the non-interactive property of the pre-shared keys, but maintain the bandwidth and memory savings by using certificateless keys. Note that pairwise secret keys can be derived in PKIs too, e.g. static Diffie-Hellman keys. However, those keys require the authentic exchange of public keys (typically using public key certificates) and are not derived in a non-interactive fashion.

4.2.2 Challenges

After discussing the benefits, we now comment on some known problems of IBC schemes and their implications to MANETs. The special role of the KGC as a

key escrow is considered a problem in most civilian applications and is one of the main reasons that prevented IBC schemes from a wide deployment so far. The KGC is a key escrow because it derives all private keys d_i in the network (see Eq. (4.2)) and is able to compute all pairwise pre-shared keys K_{ij} using publicly available information and master key s, as shown in Eq. (4.5). We will discuss the key escrow problem in great depth in Chapter 6, where we discuss existing escrow prevention solutions and introduce methods to decrease or increase the likelihood of key escrow in MANETs.

Another drawback of IBC schemes is the requirement of a confidential and authentic channel between the KGC and each network node for the secure distribution of private keys. However, when using a blinding technique as proposed in [81], an authentic channel (such as required in PKI schemes) is sufficient. We will use such a blinding technique in our security framework to enable the use of authenticated channels for node initializations.

As in all security solutions in MANETs (secret key and public key), key revocation is difficult to provide in IBC schemes employed in MANETs due to the lack of a central TTP. However, providing revocation is essential in MANETs due to the likelihood of node compromises and malicious nodes. An approach to provide revocation in MANETs is using a distributed TTP, as in [125], or so-called accusation schemes, as in [32, 85, 124]. We present the first key revocation schemes for IBC schemes employed in MANETs that is completely self-organized and does not require any external or internal TTP in Chapter 5.

Finally, providing key renewal is difficult in IBC schemes. The public key format with embedded expiration date as given in Eq. (4.3) is only sufficient in schemes without revocation. In schemes with explicit key revocation, public keys can be revoked before they expire and thus nodes might want to instantly request new keys. However, static IDs and fixed expiry intervals prevent key renewals before the next expiry interval. We address the problem of key renewal in Chapter 5 as part of our revocation scheme.

4.2.3 Related Work

Recently, IBC schemes have been considered for securing MANETs [33, 77, 124] due to their efficient key management and other desirable features as discussed in Section 4.2.1. [33, 77] both propose emulating an internal KGC using (k, n)-threshold schemes, as previously introduced for internal CAs in PKIs [85, 125]. The key management in both solutions is entirely self-organized and the authors claim that their schemes are more efficient than fully self-organized PKIs due to the efficient key management of the underlying IBC schemes. However, threshold schemes generally introduce a lot of computational and communication overhead.

In [124], an external KGC is used to distribute all keys, whereas an internal distributed KGC provides key revocation and key renewal. This approach significantly reduces the cost of network and node initialization. While [124] provides key revocation and key renewal algorithms, the schemes in [33, 77] do not provide such algorithms. We review existing revocation schemes for MANETs in more detail and introduce our own revocation scheme in Chapter 5. So far, no ID-based authentication, key exchange and other security protocols that are especially designed for MANETs have been proposed. In [33], the authors suggest using a pre-shared key for encryption. However, static keys should not be used for encryption and key exchange protocols that establish a fresh session keys are desirable.

4.3 Basic IBC Framework for MANETs

In this section, we discuss how proper identities of all network nodes should be chosen. Then we introduce the algorithms of our basic authentication and key exchange framework for MANETs and possible extensions.

4.3.1 Choosing Identities

Suitable identities used in our ID-based framework must satisfy the following properties:

1. *unique* for each entity in the network

2. *unchangeably bound* to an entity for its entire *lifetime*

3. *not transferable*

The string of information that can be used as identity depends on the application and needs to be chosen accordingly. For example, it has to be considered who needs to be authenticated or identified in the network. Generally, we can distinguish three cases of entities an identity is bound to:

1. a *user* operating a network node

2. a *node*

3. a *network interface* of a node

In the first case the ID string corresponds to a user, e.g. his email address. In that case multiple users are able to share the same device. For example, if an application enables two users to securely communicate with each other, the use of user-dependent IDs is desirable. In the second case, the ID is bound to hardware of a device, e.g. to the MAC address. In sensor networks and other MANETs in which users do not operate the network nodes, the MAC address seems to be a good choice. A combination of both previous approaches is possible using two different sets of ID-based keys to meet the requirements of different protocol layers. In the last case, the ID corresponds to a network interface of a node and might be derived from an IP address. However, we cannot generally assume network addresses such as IP addresses in all MANETs, because nodes are mobile and may join or leave the network at any time.

4.3.2 Basic Framework

We now introduce our ID-based authentication and key exchange framework. The framework is based on the BF scheme and is suitable for MANETs with external KGCs. The external KGC sets up all nodes with their private keys before the nodes join the network. This corresponds to TTP availability scenario AV-2 in Fig. 2.2 as discussed in Section 2.1.6. Once, in the network, nodes cannot access the external

KGC any longer. During network and node set up, the external KGC can be utilized to execute algorithms for system and node set up, whereas in the running network all algorithms are executed in a self-organized manner by the network nodes. The basic framework is specified by 4 algorithms: *(1.) Setup, (2.) Extract, (3.) Distribute*, and *(4.) Pre-Authentication*. Algorithms 1-3 are executed by an external KGC, i.e. outside the network during network or node initialization. These algorithms are executed by the KGC to initialize nodes before they join the network. Algorithm 4 is executed by the network nodes, i.e. completely independent of any KGC. These four algorithms constitute a basic framework that provides secure and efficient pre-authentication between pairs of nodes and enables nodes to securely execute other security protocols. As an example of such security protocols, we present a set of ID-based AKE protocols in Section 4.4 that can be seamlessly integrated into our security framework. We introduce the four algorithms of the basic framework in the following paragraphs and discuss extensions in Section 4.3.3.

(1.) Setup. On the input of a security-parameter k, the KGC selects two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order q, where q is a prime, and a map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$. The map $\hat{e}$ is bilinear, non-degenerate, and computable, and the parameters are chosen such that the BDH problem is hard in $\mathbb{G}_1$, as defined in Section 4.1. Furthermore, the KGC chooses a random generator $P \in \mathbb{G}_1$, picks a random number $s \in \mathbb{Z}_q^*$ and computes $P_{pub} = sP$. The parameters (s, P_{pub}) are the KGC's long-term private and public key. In addition, the KGC selects two hash functions $H_1 : \{0,1\}^* \mapsto \mathbb{G}_1^*$ and $H_2 : \mathbb{G}_2 \mapsto \mathbb{Z}_q^*$. First is used to derive nodes' public keys from their binary identity strings and latter to generate blinding factors for secure key distribution. After the set-up is completed, the KGC makes the following system parameters publicly available $params = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, P_{pub}, H_1, H_2 \rangle$. The KGC's long-term private key s, also referred to as master key, is kept confidential.

(2.) Extract. The KGC extracts the long-term private key d_i for each network node i with identity $ID_i \in \{0,1\}^*$. For doing so the KGC first derives the node's public key $Q_i = H_1(ID_i||t_x)$ according to Eq. (4.3) and then computes the private key $d_i = sQ_i$ using the system's master key.

(3.) Distribute. During private key distribution, the KGC bootstraps all nodes with their private keys d_i. Our security framework provides two ways of private key distribution, the first one (Scenario A) requires an authentic and confidential communication channel between KGC and each network node, whereas the second case (Scenario B) requires only an authentic channels.

Scenario A. Upon successful authentication of node i to the KGC, the KGC sends the private key d_i to i over a secure channel. The channel needs to be confidential and authentic. For example, such a channel is established if users physically go to the KGC or if private keys are directly embedded in the node during manufacturing. Node i verifies its private key by checking whether the following equation

$$\hat{e}(d_i, P) = \hat{e}(Q_i, P_{pub}) \tag{4.6}$$

is true.

Scenario B. The condition for the distribution channel can be relaxed in order to enable secure key distribution in a wider field of MANET applications. This can be done by using a simple blinding technique to protect the confidentiality of the private keys similar to the method proposed in [81]. Node i chooses a blinding factor x, computes $X = xP$, then sends (ID_i, X) to the KGC over an authentic channel. The KGC computes a blinded private key $d_i' = H_2(\hat{e}(sX, P_{pub}))d_i$ and sends it over an authentic channel back to ID_i. Node ID_i derives its private key d_i by removing the blinding factor, i.e.

$$d_i = \frac{d_i'}{H_2(\hat{e}(P_{pub}, P_{pub})^x))}. \tag{4.7}$$

Somebody who is listening to the key distribution channel cannot derive the private keys because the blinding factor can only be removed by node i and the KGC with $H_2(\hat{e}(P_{pub}, P_{pub})^x)) = H_2(\hat{e}(sX, P_{pub}))$. Node i verifies its private key d_i by checking whether Eq. (4.6) is true.

(4.) Pre-Authentication. Whenever two nodes i and j wish to communicate for the first time, they each compute their pairwise pre-shared key K_{ij} according to Eq. (4.4). Alternatively, the computations can be done at once for all potential

communication partners. The computation is non-interactive and no messages or keys need to be exchanged in this step. After their computations, the keys can be stored for future communications with the same nodes.

4.3.3 Enhancements

After introducing the basic framework, we now point out some possible extensions. We propose solutions to some of these extensions in the following sections and chapters.

- *ID-based Security Protocols.* The proposed framework establishes pairwise pre-shared keys that can be used in ID-based security protocols to achieve more security objectives. For instance, the pre-shared keys can be used to derive symmetric keys to be used in secure routing protocols that employ symmetric cryptography and thus require pre-shared secrets, e.g. [63,65,100]. The pre-shared keys can be used in any symmetric authentication and key exchange protocol and we introduce a set of ID-based AKE protocols that utilize the pre-shared keys in Section 4.4. The proposed protocols offer a variety of security properties and can be seamlessly integrated into the proposed framework .

- *Key Revocation and Key Renewal.* The framework can be extended to provide key revocation and key renewal schemes which are both essential mechanisms in MANETs due to the likelihood of node compromises. Keys must be revoked in the running network and thus the revocation algorithm must be executed in a self-organized fashion. If revocation is offered, nodes must be able to request a new key upon revocation of their current key, i.e. an algorithm for key renewal is needed. We introduce a novel completely self-organized key revocation scheme for MANETs that can be used as extension to the introduced framework in Chapter 5. The revocation scheme is complemented by a key renewal scheme presented in the same chapter.

  TTP Availabilities. The basic framework works for TTP availability scenario AV-2 as illustrated in Figure 2.2. Hence, possible extensions to the basic

framework could be adaptations to other availability scenarios. MANET applications without the support of any external TTP, i.e. AV-4, require all algorithms to be executed in a self-organized fashion by the network nodes themselves. Therefore, Algorithms 1-3 in the basic framework must be executed by an internal KGC. Such an internal KGC can be implemented using a (k, n)-threshold scheme. These kind of distributed KGCs have been presented in [33, 77, 124] for pairing-based IBC schemes employed in MANETs and the *Setup, Extract*, and *Distribute algorithms* can be adopted from these solutions. Note that Algorithm 4 *Pre-authentication* is fully self-organized in the basic framework and thus does not require any modifications for a fully self-organized solution.

For MANET applications with TTP availability AV-3, an external TTP can implement and execute Algorithms 1-3 from the basic framework to initialize all nodes that are present at network initialization. The in the running network a distributed (k, n)-KGC executes these algorithms to initialize all subsequently joining nodes, as described in the previous paragraph. Again, Algorithm 4 does not need to be modified.

Finally, in MANET applications with TTP availability AV-1, the basic framework could be directly implemented without any changes. However, the proposed Algorithms 1-3 could be optimized to take advantage of (sporadic) backbone access. Even though the basic algorithms cannot be significantly optimized, possible extensions to the basic framework can be significantly improved by utilizing sporadic network/backbone access. We discuss benefits and challenges of such MANET applications and their impact on the design of security solutions, including our proposed security framework, in Chapter 7.

4.4 ID-Based AKE Protocols

We are now introducing a set of ID-based AKE protocols that is suitable for implementation in MANETs and our proposed security framework. All AKE protocols presented in this section are designed to meet the design goals specified in Sec-

tion 3.3 and achieve all necessary and some of the desirable security properties defined in Section 2.2. We first present a lightweight protocol, Protocol 5. Then, we gradually add more security features resulting into more complex Protocols 6-10. Please refer to Table 5.1 for notations. An early version of our protocols can be found in [53].

Protocol 5. Lightweight ID-based AKE Protocol

PRE-SHARED KEY: $(k_a, k_d) = (f_{K_{ij}}(1), f_{K_{ij}}(2))$

PROTOCOL FLOW:

1. $i \longrightarrow j : ID_i, SID, N_i$
2. $i \longleftarrow j : j, SID, N_j, r_j = h_{k_a}(ID_i, N_i, SID, N_j)$
3. $i \longrightarrow j : ID_i, SID, r_i = h_{k_a}(ID_j, N_j, SID, N_i)$

SESSION KEY: $SK = f_{k_d}(N_i, N_j)$

4.4.1 Protocols with Pre-Shared Keys

Protocol 5: We introduce a lightweight ID-based AKE protocol which utilizes the pre-shared key K_{ij} from Eq. (4.4). The protocol is based on a MAC-based AKE protocol using symmetric session key derivation as discussed in Protocol 2 and the message flows are illustrated in Protocol 5. The pre-shared key K_{ij} is used to derive an authentication key $k_a = f_{K_{ij}}(1)$ and a key derivation key $k_d = f_{K_{ij}}(2)$ employing a secure KDF $f_k(\cdot)$. The key k_a is used as input to a MAC function $h_k(\cdot)$ to provide mutual authentication, whereas k_d is used as input in a KDF $f_k(\cdot, \cdot)$ to derive the session key SK.

In the first step, party i randomly chooses a nonce $N_i \in \{0,1\}^{2k}$, where k is a security parameter, and sends (ID_i, SID, N_i) to j, where SID is a session identifier. Upon receipt of the message, j randomly chooses a nonce $N_j \in \{0,1\}^{2k}$, computes $r_j = h_{k_a}(ID_i, N_i, SID, N_j)$ and sends (ID_j, SID, N_j, r_j) to i. Upon receipt, i verifies r_j and (if successful) i computes $r_i = h_{k_a}(ID_j, N_j, SID, N_i)$,

sends (ID_i, SID, r_i) to j and computes the session key $SK = f_{k_d}(N_i, N_j)$. Upon receipt, j verifies r_i and if successful computes $SK = f_{k_d}(N_i, N_j)$.

Protocol 6. ID-based AKE Protocol Using Signatures

PRE-SHARED KEY: K_{ij}

PROTOCOL FLOW:

1. $i \longrightarrow j : ID_i, SID, N_i$
2. $i \longleftarrow j : ID_j, SID, N_j, S_{d_j}(ID_j, SID, N_j, N_i, ID_i)$
3. $i \longrightarrow j : ID_i, SID, S_{d_i}(ID_i, SID, N_i, N_j, ID_j)$

SESSION KEY: $SK = f_{K_{ij}}(N_i, N_j)$

We now discuss the security properties of Protocol 5:

NP-1: Nonces N_i and N_j act as challenges, where the responses are computed as MACs over the challenges using pre-shared key k_a. Since the KGC is a key escrow, this property only holds if the KGC is neither malicious nor compromised.

NP-2: Pre-shared key K_{ij} and thus k_d are only known to i and j, which follows that SK can only be computed by these parties. Again, the KGC is able to compute all pre-shared keys (see Eq. (4.5)) and thus the property only holds if the KGC is honest and not compromised.

NP-3: The session key is derived in a symmetric fashion, thus both parties compute the session key $SK = f_{k_d}(N_i, N_j)$ in the same way.

DP-1: All session keys SK are computed independently from each other and cannot reveal any information of other secret keys.

DP-2: Pre-shared keys k_a and k_d are derived from the identity of the respective communication partner. Thus both principals can be sure about the identity of their communication partner.

DP-3: Both principals contribute their own fresh input to the session key computation. In order to achieve strict key control, derivation function $f_{k_d}(\cdot)$ has to be chosen in a way that j cannot effectively manipulate the outcome of the session key computation by choosing some special values of his key share N_j.

DP-4: Deniability is provided because pair-wise shared keys are known to a pair of principals. Hence, j can deny to have talked to i, because i could have simulated a protocol run without interacting with j.

Protocol 7. ID-based AKE Protocol Using ECDH

PRE-SHARED KEY: K_{ij}

PROTOCOL FLOW:

1. $i \longrightarrow j : ID_i, SID, T_i$
2. $i \longleftarrow j : ID_j, SID, T_j, h_{K_{ij}}(ID_i, T_i, SID, T_j)$
3. $i \longrightarrow j : ID_i, SID, h_{K_{ij}}(ID_j, T_j, SID, T_i)$

SESSION KEY: $SK = r_i T_j = r_j T_i$

Protocol 6: We now present Protocol 6 which is derived from Protocol 5 by replacing the MAC function with an ID-based signature scheme $S_{d_i}(\cdot)$. The protocol is based on signature-based AKE protocol such as illustrated in Protocol 3 and discussed in Section 2.2.3. Protocol 6 achieves the same security properties as Protocol 5, except Protocol 6 achieves KCI resilience (DP-5) but cannot provide deniability (DP-4) due to the use of digital signatures. An implementation of Protocol 6 in the proposed authentication and key exchange framework requires two additional algorithms, namely Algorithms *5. Sign* and *6. Verify*, where any pairing-based ID-based signature scheme can be used for that matter, e.g. [51].

Protocol 7: To provide perfect forward secrecy (PFS), we now present Protocol 7 which replaces the symmetric session key computation in Protocol 5 with an EC-DH key agreement, i.e. $SK = r_i T_j = r_i r_j P = r_j T_i$. Using rather EC-DH than

finite field DH is desirable because the underlying ID-based scheme already utilizes elliptic curves. The protocol is a MAC-based AKE protocol with public key session key derivation, as illustrated in Protocol 2 and discussed in Section 2.2.3. Protocol 7 preserves all security properties of Protocol 5 and additionally provides PFS (DP-6) and TTP-PFS (DP-7). PFS is achieved by using EC-DH, i.e. the ephemeral private keys r_i and r_j used for session key computation are not a part of the exchanged messages. Similarly, TTP-PFS is achieved because even if an adversary knows the master key s, he does not know r_i and r_j and thus cannot derive the session key. An implementation of this protocol in the proposed authentication and key exchange framework requires an additional Algorithm *7. EC-DH*.

4.4.2 Protocols without Pre-Shared Keys

In this section we present three ID-based AKE protocols that provide more security features than the previous Protocol 5-7 for sacrificing computational and communication efficiency. These protocols use either ID-based signatures or public key encryption to provide authentication and/or an EC-DH key agreement to derive the session key. Hence, Protocol 8-10 do not require pre-shared keys.

Protocol 8. ID-based AKE Protocol Using Signatures and EC-DH

Protocol Flow:

1. $i \longrightarrow j : ID_i, SID, T_i$
2. $i \longleftarrow j : ID_j, SID, T_j, S_{d_j}(ID_j, SID, T_j, T_i, ID_i)$
3. $i \longrightarrow j : ID_i, SID, S_{d_i}(ID_i, SID, T_i, T_j, ID_j)$

Session Key: $SK = r_i T_j = r_j T_i$

Protocol 8: Combining Protocols 6 and 7, i.e. using digital signatures for authentication and EC-DH key agreement for session key derivation, yields Protocol 8 that achieves KCI resilience (DP-5) and all security properties of Protocol 7 except of

deniability (DP-4). Again, using digital signatures prevents KCI attacks but prevents principals from denying their participation in a protocol run. Implementing this protocol in the basic framework requires the implementation of Algorithms 5, 6, and 7.

Protocol 9. ID-based AKE Protocol Using Public Key Encryption

---

PROTOCOL FLOW:

1. $i \longrightarrow j : ID_i, SID, E_{Q_j}\{N_i\}$
2. $i \longleftarrow j : ID_j, SID, E_{Q_i}\{N_j\}, h_{N_i}(ID_i, SID, E_{Q_i}\{N_j\})$
3. $i \longrightarrow j : ID_i, SID, h_{N_j}(ID_j, SID, E_{Q_j}\{N_i\})$

SESSION KEY: $SK = f(N_i, N_j)$

Protocol 9: Protocols 5-8 provide either deniability (DP-4) or KCI resilience (DP-5), but fail to provide both security features at once. We show that when applying public key encryption, we can design protocols that are resistant to KCI attacks and provide the deniability feature at the same time, which cannot be achieved by using signatures or MACs. Protocol 9 can be obtained from Protocol 5 by encrypting the exchanged nonces N_i and N_j with an ID-based public key encryption scheme $E_{Q_i}\{\cdot\}$ under the receiver's public key Q_i. Both parties decrypt the received challenges using their private key. The decrypted challenge is then used as a MAC key. The session key $SK = f(N_A, N_B)$ is derived from the exchanged nonces. Protocol 9 preserves all security properties of Protocol 5 and additionally provides DP-5 and DP-6. DP-5 is achieved by forcing the receiver of a message to use its private key to decrypt and DP-4 is achieved because no messages are signed. Protocol 9 provides *partial forward secrecy*, because an adversary needs to know both private keys d_i and d_j in order to decrypt the exchanged messages and obtain the session key. Hence, the knowledge of one private key is not sufficient for an attack. Note that an implementation of Protocol 9 in the basic framework requires two additional algorithms, namely *8. Encryption* and *9. Decryption*, e.g. using the BF scheme.

Protocol 10: In order to develop a protocol that resists KCI attacks, achieves deniability, and provides PFS, public key encryption can be combined with an EC-DH key agreement. Hence, we combine Protocol 7 with Protocol 9 to derive Protocol 10. The protocol inherits all properties from Protocol 9 and additionally provides PFS and TTP-PFS.

Protocol 10. ID-based AKE Protocol Using Public Key and EC-DH

Protocol Flow:

1. $i \longrightarrow j : ID_i, SID, E_{Q_j}\{T_i\}$
2. $i \longleftarrow j : ID_j, SID, E_{Q_i}\{T_j\}, h_{T_i}(ID_i, SID, E_{Q_i}\{T_j\})$
3. $i \longrightarrow j : ID_i, SID, h_{T_j}(ID_j, SID, E_{Q_j}\{T_A\})$

Session Key: $SK = r_i T_j = r_j T_i$

4.5 Security Analysis

We now analyze the security of the previously proposed authentication and key exchange framework and ID-based AKE protocols.

4.5.1 Authentication and Key Exchange Framework

The proposed ID-based authentication and key exchange framework employ the BF scheme as underlying IBC scheme and the first three algorithms of the framework are adopted from the BF scheme. When the IBC scheme is set up according to the conditions specified in the respective algorithms, the BF scheme has been shown to be IND-ID-CCA secure [17], i.e. to provide chosen ciphertext security. Note that if vulnerabilities of the BF scheme will be discovered in the future, the underlying IBC scheme in our security framework can be replaced by any other secure pairing-based IBC scheme. Scenario B in Algorithm 3 provides secure key distribution, because the blinding factor can only be computed by node i and the KGC (see Eq. (4.7)).

Algorithm 4 does not require any message exchanges and thus cannot be attacked by any eavesdropper or active attacker, where the pre-shared keys K_{ij} have been shown to be secure in [17, 19]. Recall that all private as well as pre-shared keys are known to the KGC. Attacks by malicious KGCs will be analyzed in Chapter 6 and the discussed countermeasures to prevent such attacks can be adopted to our security framework.

The fully self-organized ID-based AKE framework as outlined in Section 4.3.3, requires collaborative computations by a group of network nodes that form the internal KGC using a (k, n)-threshold scheme. The security of those collaborative computations in Algorithms 1-3 depend on the underlying ID-based threshold schemes [6, 17, 33, 77].

4.5.2 ID-based AKE Protocols

We already discussed the provided security properties of the presented ID-based AKE protocols. For an easier comparison, Table 4.1 summarizes the desirable security properties (DP) of all protocols. For completeness we also list non-repudiation (DP-8) and replay resilience (DP-9). Note that non-repudiation is provided by all protocols employing digital signatures, where the property is by definition mutually exclusive with deniability. Replay resilience is provided by all presented protocols because they all employ nonces using a challenge-response technique. We can observe that all presented protocols provide all necessary properties NP1-NP3 and desirable properties DP1-DP3 and DP-9. As a consequence, the protocols resist the most common attacks, such as impersonation, known session key, UKS, and replay attacks. The presented set of AKE protocols demonstrate the typical security-performance trade-off of cryptographic security protocols, because protocols that offer more features to thwart additional attacks induce larger computational and communication overhead.

PFS is not achieved in Protocols 5 and 6, because they use very efficient symmetric key derivation functions rather than computationally demanding DH key agreements. However, the protocols are still attractive for many MANET applications because PFS is not necessary in some scenarios, especially in all applications

in which authenticity of data is more important than secrecy of the data, as discussed in [79]. Partial FS, as provided in Protocol 9, is desirable if parties have different roles, such as in server-client scenarios in which compromised servers are far less likely than compromised clients. However, for most MANET application we assume all nodes to have equal roles and thus their private keys are equally strong protected. This fact makes partial FS less interesting for MANET applications.

Protocols 9 and 10 are the only protocols that achieve KCI resilience (DP-5) and deniability (DP-4). Both protocols employ public-key encryption to enable achieving both properties at the same time. Interestingly, Boyd, Mao and Paterson suggested that deniability and KCI resilience might be mutually exclusive [19]. Note, that another protocol that achieves both properties by applying the discussed principal is the SKEME protocol [79].

In addition to analyzing the security properties of the protocols, we now point out how the protocols can be formally analyzed in the Canetti and Krawczyk security model for key exchange protocols [25]. The model helps to analyze the security of n-party message driven key exchange protocols with point-to-point communication. In their model the security of a protocol Π is first shown to be secure in an authenticated-link model (AM) in which an adversary can only passively eavesdrop on the communications. The protocol is then transformed into a protocol Π' that is secure in the more realistic unauthenticated-link model (UM) by using so-called *authenticators*. We omit a detailed description of the formal security model and refer the interested reader to the original papers [10, 25]. The ID-based AKE protocols presented in Section 4.4 are all designed using authenticators, namely MAC-based authenticators [25], signature-based authenticators [10], and encryption-based authenticators [10]. Instead of providing a formal security proof, we describe how the presented protocols can be derived from existing protocols that are proven secure in the model. Furthermore, we list the conditions for each protocol to be secure.

Protocol 5: Protocol 5 is similar to a *REKEY* protocol in [25] in which pre-shared keys K_{ij} are used. In [19], pre-shared keys from Eq. (4.4) are shown to be secure for their use in MAC-based authenticators. This proof combined with the security proof of the *REKEY* protocol in [25], allows us to conclude that Protocol 5 is secure

without PFS in the UM model if the following three conditions hold:

Condition 1. Pre-shared keys K_{ij} from Eq. (4.4) are random keys chosen under security parameter k.
Condition 2. The BDHP is hard in the implemented IBC scheme.
Condition 3. The employed MAC function $h_k(\cdot)$ is secure.

Protocol 6: Protocol 6 is derived from Protocol 5 by replacing the MAC-based authenticator by signature-based authenticators [10]. Hence, Protocol 6 is secure without PFS in the UM model if Conditions 1, 2 and the following Condition 4 hold:
Condition 4. The employed ID-based signature scheme $S_d(\cdot)$ is secure against chosen message attacks.
Protocol 7: The protocol employs an EC-DH key agreement in which mutual authentication is provided by MAC-authenticators. Similar to the *SIG-DH* protocol in [25] that applies signature-based authenticators to a DH key agreement, Protocol 7 is secure with PFS in the UM model if Conditions 1-3 and the following Condition 5 hold:
Condition 5. The ECDH problem must be hard in the implemented IBC scheme.

Note that a similar protocol was proven secure in the UM model in [19].

Protocol 8: The protocol is identical to the *SIG-DH* protocol in [25], except that a EC-DH key agreement is used rather than a finite field one. Hence, Protocol 8 is secure with PFS in the UM model if Conditions 4 and 5 hold.
Protocol 9: The protocols employs public key encryption-based authenticators [10] replacing the MAC-based authenticators in Protocol 5. The protocol is secure without PFS in the UM if the following two conditions hold:
Condition 6. The employed ID-based encryption scheme $E_Q\{\}$ is secure against chosen cipher text attacks (CCA-secure) and MAC function $h_k(\cdot)$ is secure.

An example of such CCA-secure IBE scheme is [24].

Condition 7. The employed key derivation function $f(\cdot)$ is secure.
Protocol 10: The protocol secures a EC-DH key agreement by using public key encryption-based authenticators [10]. Hence, the protocol is secure in the UM with PFS if Conditions 5 and 6 hold.

Protocol	Desirable Properties (DP)								
	DP-1	DP-2	DP-3	DP-4	DP-5	DP-6	DP-7	DP-8	DP-9
Protocol 5	X*	X	X	X	–†	–	–	–	X
Protocol 6	X	X	X	–	X	–	–	X	X
Protocol 7	X	X	X	X	–	X	X	–	X
Protocol 8	X	X	X	–	X	X	X	X	X
Protocol 9	X	X	X	X	X	o‡	–	–	X
Protocol 10	X	X	X	X	X	X	X	–	X

Table 4.1: Desirable Security Properties of Protocols 5-10

*X denotes that the security property is provided by the protocol
†– denotes that the security property is not provided by the protocol
‡o denotes that the security property is partially provided by the protocol

4.6 Performance Analysis

We now analyze the performance of the proposed ID-based authentication and key exchange framework and ID-based AKE protocols with respect to memory requirements, communication and computational overhead which represent the main constraints in MANETs.

4.6.1 Authentication and Key Exchange Framework

Efficient implementations of IBC schemes and bilinear parings have been introduced in the literature [9, 13, 17, 92]. To achieve 1024-bit security and an efficient implementation, a 512 bit curve is chosen, where computations are executed in 170 bit subgroups. This follows that keys and pre-shared keys are 170 bits long. Furthermore, we believe identities ID_i of size 64 bits are sufficient, e.g. 32 bits static data and 32 bits for dynamic data such as expiry date t_x. Consequently, storage requirements are very low in the presented scheme and can compete with EC-based PKI implementations. Nodes can either store pre-shared keys and/or public keys together with the corresponding identities or derive these keys on-demand. Computing pre-shared keys requires a pairing computation and public keys the use of mapping function $H_1(\cdot)$. Storing versus on-demand computations consti-

tutes a memory/computation trade-off and the best implementation needs to be chosen according to the nodes' constraints in particular applications. With small key sizes and growing memory space, storing keys seems more desirable in most applications. In addition to low memory requirements, the employed IBC scheme reduces the bandwidth requirements because neither keys nor certificates need to be exchanged.

The computational complexity of the proposed security framework depends on the implemented bilinear pairing, the number and frequency pre-shared keys need to be computed, as well as the implemented ID-based AKE protocol. The performance of the ID-based AKE protocols is discussed in more detail in the next section. Computing pre-shared keys K_{ij} as well as Protocols 6, 8, 9 and 10 require the computation of bilinear pairings, such as Weil or Tate pairings. Latter pairing is favored due to its better computational performance. Clearly, the pairing computation is the most demanding computation in framework and protocols. However, efficient algorithms and implementations exist, e.g. [9, 13, 92] and the Tate pairing has been implemented on such constrained platforms as smartcards [13]. Nevertheless, the number of required pairing computations should be limited, because on battery-operated devices, such as cell phones or PDAs, demanding computations can drain the battery. In very constrained environments in which potential communication partners are known ahead of time, the one-time computation of pre-shared keys can be performed off-line. For instance, K_{ij} can be computed on a desktop computer, and then be downloaded to a user's PDA. In another scenario, a central TTP computes pairwise shared keys of all network nodes and distributes them securely to each node at the time when nodes are bootstrapped with their private keys. Note that the KGC is able to compute all pre-shared keys in any IBC scheme and thus the previous scenario does not give the KGC additional power. In the extreme case that communication partners are not known in advance and the nodes cannot compute pairings, node i could send a communication request to j, where j requests and downloads the pre-shared key K_{ij} next time it is docked to a more powerful device or connected to a TTP.

The overall network performance, especially with respect to computational and

communication costs highly depends on the implementation of an external or distributed internal KGC. Latter, fully self-organized implementation, shows significantly worse performance due to the use of (k, n)-threshold schemes. Hence, this implementation should only be used when an external TTP is not available at all (AV-4). In all other scenarios, nodes should be at least initialized by an external TTP and should the external TTP no longer available, all subsequent initializations can be carried out by an internal KGC (AV-3). The initialization by an external TTP is feasible in our target applications specified in Section 3.3. Furthermore, we believe that this type of TTP availability occurs in many real-world MANET applications. For example, network nodes may be initialized by their manufacturer, network provider, service provider or system administrator.

4.6.2 ID-based AKE Protocols

To achieve good performance in terms of communication costs, all presented ID-based AKE protocols have three protocol flows which is the minimum number of flows to provide mutual entity authentication using nonces [89]. For computational efficiency, symmetric cryptographic primitives are used whenever possible and nonces are used for two purposes: (1) as challenges for entity authentication, and (2) as fresh inputs for session key derivation. We now analyze the performances of the individual protocols.

We can observe that Protocol 5 uses only symmetric primitives except of the computation of pre-shared keys K_{ij}. However, the pre-shared keys only need to be computed once and can be stored for future communications with the same node. In addition, the computation can be performed off-line, i.e. before the protocol execution, as discussed in the previous section. No interaction between the communicating parties is required to share keys k_A and k_d because both are derived from the non-interactively pre-shared key K_{ij}. We conclude that Protocol 5 shows excellent computational and communications performance, and is thus well-suited for MANET applications in very constrained environments. All other presented protocols are less efficient, because they require the on-line computation of some pairings and modular exponentiations (Protocol 6, 7, 9 and 10) and/or scalar mul-

tiplications (Protocol 8, 9, and 10). These computations are orders of magnitude more demanding than symmetric operations.

In particular, Protocols 6 and 8 both employ signatures and the protocol performance depends on the implemented ID-based signature scheme. For instance in [51], signing takes two modular exponentiations in $\mathbb{G}_1$ and verifying takes two pairing computations and one modular exponentiation in $\mathbb{F}_p$. Protocols 7, 8 and 10 require the implementation of EC-DH, which requires the computation of two scalar multiplications in $\mathbb{G}_1$ per node. Protocol 9 and 10 employ an ID-based encryption scheme. In the BF scheme, encryption takes one pairing computation and decryption one pairing and one modular exponentiation in $\mathbb{G}_1$. Please note that in order for Protocols 9 and 10 to be secure, the encryption scheme needs to be CCA-secure which is not provided by the original BF scheme. A CCA-secure ID-based encryption scheme is introduced in [24]. However, CCA security adds a lot of additional overhead which makes it undesirable for implementations in MANETs. Protocols 9 and 10 are the only protocols providing deniability and KCI resilience at the same time. Hence, the protocols are of interest whenever both features are required and should be replaced by a more efficient protocol whenever one of the feature is sufficient.

4.7 Discussions and Conclusions

In this chapter, we proposed an ID-based authentication and key exchange framework that enables efficient and secure pre-authentication, authentication and authenticated key exchange among network nodes in MANETs. The basic scheme is suitable for MANET applications with KGC availabilities AV-2 (see Figure 2.2), and as part of our discussed extensions in Section 4.3.3, we outlined how the framework can be adopted to KGC availabilities AV-3 and AV-4. Our framework is flexible and can be implemented with any pairing-based IBC scheme, where the security of the framework is based on the implemented IBC scheme. We described an algorithm for efficient system set up in which costs are solely carried by an external KGC. In addition, pre-authentication is efficient and secure because pairwise

pre-shared keys are derived in a non-interactive fashion. The derived pairwise pre-shared keys enable the use of any symmetric or ID-based AKE protocol to establish fresh session keys.

The presented set of ID-based AKE protocols in this chapter can be seamlessly integrated in our proposed authentication and key exchange framework but also serve as an independent solution. The first AKE protocol is an extremely efficient and purely symmetric protocol that utilizes pairwise pairing-based keys as pre-shared secrets. We then derived more protocols by gradually adding security features, which sacrifices some of the computational efficiency of the first protocol. In our security analysis we prove which security properties are achieved by each protocol. Our performance and security analysis enables the selection of the most efficient AKE protocol for particular MANET applications depending on network constraints and security needs.

Chapter 5

Self-Organized Key Revocation for MANETs

Many proposed PKI and IBC schemes for MANETs do not provide schemes for key revocation and key renewal. However, due to the weak physical protection of nodes combined with node exposure in potentially hostile environments, node compromises including key disclosures are very likely in MANETs. Hence, we believe that key revocation and key renewal are of great importance in MANETs and every node should be able to instantly verify whether a public key has been revoked. Frequent key renewals to prevent key compromises are either computationally challenging in solutions with distributed on-line TTPs or simply infeasible in solutions with off-line TTPs. In this chapter, we propose key revocation and key renewal schemes for IBC schemes that are especially designed to meet the requirements and constraints of MANETs. The proposed schemes can be seamlessly integrated in the ID-based authentication and key exchange framework from Chapter 4. In addition, the schemes can be used in any pairing-based IBC scheme and the revocation scheme can be modified to serve as certificate revocation scheme in PKI solutions in MANETs.

In our revocation scheme, each node uses a neighborhood watch algorithm to monitor nodes in communication range for suspicious behavior. All observations are securely propagated to an m-hop neighborhood. The public key of a node is

revoked if at least δ nodes accused that node. Our key revocation scheme is scalable in parameters m and δ, i.e. the level of security can be chosen as performance trade-off. To enable key renewal in IBC schemes, we introduce a new format for ID-based public keys such that new keys can be issued for the same identity after the previous key has been revoked. In addition, we discuss and efficiently solve two problems of nodes wishing: 1) to revoke their own keys and 2) to learn about past accusations and revocations upon joining the network.

The remainder of this chapter is organized as follows. In the next section we summarize some previous work on revocation and monitoring in MANETs and point out differences to our schemes. In Section 5.2, we discuss the system set-up for our key revocation and key renewal schemes and introduce the schemes in Section 5.3. The security and performance of the proposed schemes are analyzed in Sections 5.4 and 5.5, respectively. Finally, we discuss the contributions of the proposed schemes in Section 5.6. An earlier version of our revocation scheme without extensions and extensive security analysis appeared in [61].

5.1 Related Work

In this section, we review some existing key revocation schemes for IBC-based solutions as well as certificate revocation schemes for PKI solutions employed in MANETs. In the second subsection, we review some monitoring schemes that have been proposed to identify and ideally exclude malicious network nodes in MANETs. Monitoring nodes is necessary in our and other existing accusation-based revocation schemes for deciding which keys should be revoked.

5.1.1 Revocation in MANETs

We discussed some general approaches of certificate revocation for traditional infrastructure networks in Section 2.2.3, e.g. CRL and OCSP. However, these widely deployed solutions are not suitable for MANETs, because they require nodes to access TTPs or on-line repositories to download or request the status of certificates. Providing key revocation is crucial in MANETs due to the likelihood of

node compromises in these networks. In this section, we will discuss previous work on revocation schemes that have been especially designed for MANETs. However, despite the importance of key revocation, several of the PKI and IBC-based schemes that have been introduced for MANETs , e.g. [31, 32, 33, 67, 77, 85, 124, 125], either completely ignore key revocation and/or key renewal or just briefly outline possible solutions. Only a few more sophisticated revocation schemes have been proposed for MANETs [32, 124].

In [125], it is suggested that a distributed on-line CA collaboratively revokes certificates. However no revocation scheme is introduced. In fact, a revocation scheme in this solution would require threshold signatures, which are computationally demanding. In [85], an accusation scheme is briefly outlined in a single paragraph. The authors propose that each node observes their neighboring nodes for malicious behavior and based on their observations, nodes propagate signed accusations to an m-hop neighborhood. All receivers verify the accusations and update their accusation lists accordingly. If the number of accusations against one node is greater than a threshold δ, this node's certificate is revoked. The problem of newly joining nodes is not addressed in [85] and would require joining nodes to verify accusation tables from its neighbors to learn about past accusations and revocations. This approach requires the verification of all previously issued accusation values received from the neighboring nodes. Even in scenarios with a moderate number of accusations, the verification process is computationally too demanding. Furthermore, an algorithm for nodes that want to revoke their own compromised keys is not proposed. In summary, [85] outlines some ideas how to provide self-organized key revocation in MANETs, but the authors neither propose schemes nor address more subtle problems.

To our best knowledge, the first paper completely dedicated to certificate revocation in MANETs is [32]. Here, the authors assume that an off-line CA issues certificates to all network nodes before they join the network. The proposed certificate revocation scheme employs an accusation scheme with threshold δ, and as in [85], certificates are revoked if the sum of received accusations against the same node is greater than δ. All accusations are frequently broadcasted throughout the

entire network. Here, each accusation has an associated weight which is a real number in the range $[0, 1]$, where the weight is computed according to the number of accusations a node has made so far, how many accusations were reported against this node, etc.. When a new node joins the network, the node receives the accusation tables from all network nodes. The accusation messages in [32] are not secured at all and the authors suggest checking inconsistencies in received accusation tables. In addition, receivers only accept accusations from senders with sufficiently large trust value, where trust values are computed in a similar manner as the accusation weights. An algorithm for nodes that want to revoke their own keys is not proposed in [32].

A very radical approach for key revocations in MANETs is to revoke the keys of accusor and accused node, as proposed in [31]. Here, the accusor broadcasts a signed accusation including its own identity and the one of the accused node. The receiver verifies the message and then revokes both keys. This method prevents false accusations from malicious nodes in a simple cost efficient way without the need of threshold schemes. Thus accusations propagate very fast throughout the network without consuming much memory space, computational power or bandwidth. However, an obvious disadvantage of the scheme is, that nodes which sent out accusations must request new keys, which can be quite difficult or even infeasible in MANETs.

Besides our scheme, we are only aware of one other key revocation scheme for IBC schemes in MANETs [124]. In [124], an external off-line KGC initializes all nodes with their first private and public key pair, before nodes join the network. In the network a distributed on-line KGC consisting of n network nodes (called D-KGC) carries out key revocation and key renewals. The distributed KGC is implemented using a (k, n)-threshold scheme. Nodes monitor their neighborhood for malicious behavior and send their accusations to b assigned D-KGCs. Once a threshold of at least δ accusations is reached, a group of k D-KGCs collaborative sign a revocation messages. Each node has two pairs of keys, a static one issued by an external KGC and one that depends of the current time interval issued by the distributed on-line KGC. Keys are updated periodically by broadcasts sent

by the D-KGCs. The message can only be processed by non-revoked nodes to derive the new keys. The locations of D-KGCs are hidden using an anonymous routing protocol to prevent attacks targeted at the on-line KGC. Furthermore, the compromise of at least k D-KGCs compromises all dynamic keys but does not compromise the master key of the external KGC. Hence, the static keys of the system are still not compromised. Newly joining nodes can ask neighboring nodes for a list of previously revoked keys. However, this requires nodes to store all received signed accusations and new nodes must verify one signature for each revoked key. This can be computationally challenging for larger numbers of revoked keys. As in all other discussed schemes, there is no algorithm for nodes that wish to revoke their own keys in [124].

5.1.2 Misbehavior Detection Schemes

Employing protocols that utilize cryptographic key material and primitives, such as the ID-based AKE protocols in Section 4.4, can prevent attacks by outsiders. However due to likelihood of node compromises in MANETs, we need to be able to identify malicious nodes in the network to prevent attacks by insiders. Once identified, further actions can be taken, such as key revocation or exclusion from routing to ultimately prevent attacks by insiders. Hence, we require a metric to measure malicious behavior, a scheme to observe the specified behavior and a scheme to punish identified nodes. Due to the lack of a central TTP, identifying and excluding/punishing malicious nodes must be carried out by network nodes themselves.

We define malicious nodes as nodes that are either compromised or selfish. We assume that compromised nodes will engage in some kind of malicious activities, otherwise these nodes cannot be detected. Selfish nodes rather save their own energy than acting as router to forward other nodes' packets. Some metrics are needed to define whether a node is malicious. These metrics can be simple rules, such as number of dropped packets, or consist of complex systems such as Intrusion Detection Systems (IDS). Nodes need schemes to measure the behavior of other nodes to apply the defined metrics, e.g. a monitoring scheme that monitors neighboring nodes. Another approach for excluding selfish nodes from the network is rewarding

well behaving nodes rather than punishing malicious ones. This has the advantage that no scheme for detecting malicious nodes is required. For instance, the authors in [22] suggests the introduction of a virtual currency to reward nodes which forward packets and to charge nodes which wish to send packets.

The following metrics have been proposed for detecting malicious nodes in MANETs:

1. count number of dropped packets
2. count number of generated packets
3. check response time of certain nodes
4. wait for messages confirming each hop on a multi-hop routing path
5. use anomaly detection systems to detect unusual behavior
6. run IDS on each node to detect so-called signatures of known attacks

Metric 1 requires nodes to count the number of packets that have been received by a neighboring node and not been forwarded, even though the packets' destination address is different from the address of the monitored node. For example, in [21,88], nodes check if a packet forwarded by themselves are forwarded by the next node on the routing path. In another flavor of this metric, nodes count the number of bits received by a neighbor and compare them to the number of output bits of the same node [20]. In Metric 2, nodes count the number of generated packets of neighboring nodes, e.g. route requests, and when this number reaches a certain threshold the node is marked as malicious [14]. This metric allows to detect denial of service (DoS) attacks which typically require the generation and propagation of numerous messages. In Metric 3, nodes send probing messages to check the response time of certain nodes [29]. Metric 4 requires the employed routing protocol to be modified such that each node on the routing path sends a confirmation to the source node [82]. Metrics 5 and 6 use more complex schemes to measure malicious behavior and compare observed behaviors to previously derived normal behavior patterns or signatures of known attacks [88, 123], respectively.

We conclude that Metrics 1, 2, 5, and 6 require the monitoring of neighboring nodes as mechanism to measure specified behavior, whereas Metrics 3 and 4 obtain their measurements from received messages that are a response to an initiated process. Furthermore, we can observe that Metrics 1-4 use nodes' routing behavior as indicator of maliciousness, whereas Metrics 5 and 6 can take more complex behavior patterns into account. Metrics 3 and 4 require the modification of employed routing protocols and 4 imposes additional network load. Metrics 5 and 6 require that nodes run special software.

All metrics for identifying malicious nodes that are based on monitoring neighbors, e.g. Metrics 1, 2, 5, and 6, are suitable for our and other accusation-based revocation schemes. For instance, the monitoring schemes proposed in [14,20,21,88,123], can be used as basis for the revocation schemes in [32, 85, 124] as well as our revocation scheme. Whenever the threshold of the applied metric is reached, the node is marked as malicious and the key revocation scheme is started by sending out accusation messages. Here, the key revocation scheme represents the intrusion response or punishment mechanism for detected malicious nodes. Monitoring schemes cannot work completely accurately and there will be always some errors associated with the implemented scheme. Typically we distinguish two types of errors, namely *false positives* and *false negatives*. Here, false positives are nodes that are marked as malicious by the monitoring scheme, where in fact the nodes are good. On the other hand, false negatives are all nodes that are marked as honest by the scheme, where the nodes are in fact malicious. The two errors of monitoring schemes are denoted as false positives rate α and false negatives rate β, where α is the ratio of falsely accused nodes to all honest nodes and β is the ratio of undetected malicious nodes to all malicious nodes. Hence, $0 \leq \alpha, \beta \leq 1$, with typical values ranging from $0.01 - 0.1$.

5.1.3 Contributions of Our Key Revocation and Renewal Scheme

In this subsection, we briefly summarize the contributions of our key revocation and renewal schemes for MANETs and discuss the differences to existing schemes.

We believe that due to the lack of an infrastructure in MANETs, only network nodes themselves are able to judge whether a node has been compromised. Hence, we believe that monitoring neighbors, using one of the mechanisms described in the previous section, in combination with an accusation-based revocation scheme are the only option for providing key revocation in MANETs. Furthermore, revocation schemes for MANETs must be resilient to malicious accusations against honest nodes. We believe the approach in [31] is too radical for most applications and thus we only consider accusation-based revocation schemes with threshold δ. Hence, we compare our accusation-based revocation schemes with threshold δ to other existing schemes that use a similar approach, namely [32, 85, 124]. Our key revocation and key renewal schemes for IBC schemes employed in MANETs have the following features that distinguish them from existing schemes:

1. Accusations are cryptographically protected against impersonations and modifications without the need of digital signatures.
2. Newly joining nodes efficiently and securely obtain previous accusations.
3. Nodes can efficiently and securely revoke their own compromised keys.
4. The key revocation scheme is completely self-organized and thus independent of any external off-line or distributed on-line KGC.
5. The ID-based public keys can be renewed before the next expiry interval.

Unlike [32], our scheme protects accusations from impersonations and modifications. An inconsistency check as in [32] is not sufficient to prevent attacks by an adversary who controls the communication channel, which is a feasible attack on wireless links. Our scheme uses pre-shared secret keys from Eq. (4.4) to protect and

verify accusations and unlike [85, 124] our scheme does not require any signature scheme. This makes our scheme computationally more efficient. Another advantage of our scheme is that newly joining nodes can obtain previous accusations without the need to verify many signatures as in [85, 124]. This makes the process of joining nodes very efficient while present nodes can save memory space that would be necessary to store signatures of accusations. Our revocation scheme is the first that provides an algorithm that enables nodes to revoke their own keys after noticing that their keys have been compromised. Our proposed so-called *harakiri* algorithm is both efficient and secure. Our key revocation scheme is completely self-organized and does not rely on any off-line or on-line KGC. This has the advantages that the scheme can be implemented in any KGC availability scenario (AV1-AV4). Furthermore, no special nodes are targets of attacks, unlike the D-KGCs in [124], because all nodes are responsible for gathering accusations. Hence, our revocation scheme does not require extra protection from anonymous routing protocols such as in [124]. We believe that despite anonymous routing, the D-KGCs in [124] can still be localized by traffic analysis.

Finally, our key renewal scheme is the first for ID-based keys that allows key renewal at any time, even before the next expiry interval. Like the key renewal scheme in [124], our scheme has a renewable yet still predictable part that can be updated for each public key renewal, however in [124] this can only be done at the beginning of a new expiry date interval.

5.2 System Set Up

We assume an existing implementation of a pairing-based IBC scheme in the network. System set up, key generation, key distribution and pre-authentication are executed according to Algorithms 1-4 in the basic framework in Section 4.3.2. Only the public key format needs to be modified to allow key renewal, which we discuss in Section 5.2.2. Furthermore, we summarize our system assumptions and notations in Sections 5.2.1 and 5.2.3, respectively. We describe how each node creates its individual key revocation list (KRL) in Section 5.2.4. Finally, we define the underlying

trust model of our key revocation and key renewal scheme in Section 5.2.5.

5.2.1 System Assumptions

The network and node assumptions that are necessary for our IBC key revocation and key renewal schemes to work can be summarized as follows:

1. bidirectional communication links
2. nodes have monitoring schemes implemented
3. each node i has a unique identity ID_i
4. nodes know identities and hop-distance of their one-hop neighbors
5. nodes know identities of all nodes in their m-hop neighborhood
6. nodes obtain a private and public key pair (d_i, Q_i) from an off-line KGC prior joining the network

The first two assumptions are necessary to enable nodes to monitor their neighbor nodes in communication range, which is required in our revocation scheme. Note that bidirectional links are a common assumption in many lower-layer MANET protocols, e.g. AODV [118] and other AODV-based routing protocols. We discussed some approaches and metrics for monitoring schemes in Section 5.1.2 and assume that a suitable scheme is implemented by all network nodes. We would like to note that most monitoring schemes require nodes to be in promiscuous mode, which is also a mandatory requirement in dynamic routing protocols, e.g. AODV and DSR [75]. Assumption 3 is necessary to unambiguously identify nodes. This kind of identifiers are required for many network tasks and protocols, including the ID-based authentication and key exchange framework and ID-based AKE protocols in Chapter 4. Assumption 4 is needed, because neighbor nodes need to be unambiguously identified before they can be marked as suspicious or trustworthy in the revocation scheme. This information is usually provided by routing and other lower-layer protocols, e.g. AODV. In case the identities of neighbors are unknown,

users must first explore their neighborhood by sending *hello* messages and waiting for responses. Assumption 5 is necessary to enable nodes to decide which accusation values they must consider for updating their revocation lists, e.g. accusations from nodes that are more than m hops away are discarded. Note that nodes do not need to know the hop-distance to their m-hop neighbors, but only the identities of all nodes that are not more than m-hops away. This kind of information can be provided by routing protocols. Assumption 6 is necessary because cryptographic keys are used to provide message authentication in our revocation scheme. Here we assume an external off-line KGC distributing initial private keys d_i to all nodes i before joining the network. Furthermore, we assume that the KGC verifies each node's identity ID_i before issuing the private keys. This TTP assumption corresponds to scenario AV-2 in Section 2.1.6 and Figure 2.2. In addition, we outline a solution with distributed on-line KGC corresponding to scenarios AV-3 and AV-4 as part of the extensions in Sect. 5.3.4.

We can summarize that all system assumptions for our schemes, except Assumption 2, are quite common in MANETs, and in fact mandatory in most ad hoc routing and security protocols. Hence, the assumptions for our key revocation and key renewal schemes do not impose much additional burden to the system.

5.2.2 Public Key Format

The validity period of cryptographic keys should be limited to reduce the likelihood of compromise. As discussed earlier, an expiry date can be directly embedded in ID-based public keys, as shown in Eq. (4.3). However, this key format is only sufficient in schemes without revocation. In schemes with explicit key revocation, public keys can be revoked before they expire. As a consequence a node might wish to request a new key before the previous one expires at time t_x. However, since identities ID_i are static in IBC schemes, issuing a key for the same expiry date t_x would result into the same compromised key d_i. On the other hand, issuing new keys with a new expiry date $t'_x > t_x$ might not be feasible, because a node i could be only eligible to possess keys until t_x. Furthermore, it is desirable in IBC schemes that expiry dates are chosen in a predictable manner, e.g. in fixed intervals

ΔT. This allows the computation of valid public keys at the beginning of each new expiration interval $t_{x+1} = t_x + \Delta T$ without the exchange of these new keys. Hence, to enable immediate key renewal after key compromise, some additional data v that can be changed with every key renewal must be added to the public keys. We introduce the following key format below

$$Q_i(t_x, v_i) = H_1(ID_i||t_x||v_i), \tag{5.1}$$

where v is the version number of i's public key. For instance, upon revoking $Q_i(t_x, v_i) = H_1(ID_i||t_x||v_i)$, node i can request a new key Q'_i at time $t < t_x$, with $Q'_i(t_x, v'_i = v_i + 1) = H_1(ID_i||t_x||v_i + 1)$. Note that the version number v always starts with $v = 1$ for every new expiry date t_x and is incremented with each key renewal that occurs before t_x.

The new key format is a trade-off between user friendliness and performance. The format in Eq. (5.1) allows a node i to request new keys at any time, but requires the notification of all other nodes about the new key. Therefore, node i can either broadcast its new key Q'_i or send a message containing the new version number v'. Recall that v always starts with $v = 1$ and hence the first keys of each interval do not need to be broadcasted. Key renewal and distribution with $v > 1$ are discussed in Section 5.3.3. The best performance is achieved for public keys of format in Eq. (4.3), but then nodes have to wait until the next expiration interval to request a new key, i.e. for a maximum time of ΔT; and only receive a key if the node is still eligible to obtain keys at that time.

5.2.3 Notations

We need to introduce some notations for our schemes. A summary of notations and symbols can be found in Table 5.1. Let N denote all network nodes in the MANET, where $\Omega = |N|$ is the number of network nodes. R is the communication range of all nodes for transmitting and receiving messages and $|x - y|$ denotes the Euclidean distance between two nodes x and y. Let $N_{1,i}$ denote i's one-hop neighbors, i.e. all nodes in immediate communication range of i, with $N_{1,i} = \{j :$

Notations	
N	Set of all network nodes
$\Omega = \|N\|$	Number of all network nodes
$N_{1,i}$	Set of i's one-hop neighbors
$\sigma_i = \|N_{1,i}\|$	Number of i's one-hop neighbors
$\overline{\sigma}$	Average number of one-hop neighbors in network
$N_{m,i}$	Set of i's m-hop neighbors
$\varrho_i = \|N_{m,i}\|$	Number of i's m-hop neighbors
$\overline{\varrho}$	Average number of m-hop neighbors in network
δ	Threshold in accusation scheme
ε	Security parameter for accusations by nodes in l-hop distance, with $l > 1$
m	Propagation range of accusation and revocation messages
$\mathcal{KRL}^i(t_x)$	i's key revocation list containing accusation and revocation information for public keys with expiry date t_x of all nodes in m-hop distance
t_x	Embedded expiry date of public keys
ΔT	Fixed expiry intervals, i.e. $t_{x+1} = t_x + \Delta T$
$\|x - y\|$	Euclidian distance between two nodes x and y
α, β	False positive and false negative rates of implemented monitoring scheme
$a\|\|b$	Concatenation of two binary strings a and b
$\underline{c}_j^i$	Column vector in $\mathcal{KRL}^i$ containing received accusations from node j
$\underline{r}_j^i$	Row vector in $\mathcal{KRL}^i$ containing received accusations against node j
r_k	counter for received k-vectors
H_i	i's honest one-hop neighbors
$n_h^i = \|H_i\|$	number of i's honest one-hop neighbors
F_i	i's falsely marked honest one-hop neighbors
$n_f^i = \|F_i\|$	number of one-hop neighbors falsely marked by i
M_i	i's malicious one-hop neighbors
$n_m^i = \|M_i\|$	number of i's malicious neighbors
U_i	i's malicious undetected one-hop neighbors
$n_u^i = \|U_i\|$	number of i's undetected malicious one-hop neighbors
C	colluding nodes
$n_c = \|C\|$	number of colluding nodes
Θ_i	i's trusted one-hop neighbors

Table 5.1: List of Notations: Key Revocation and Renewal Schemes

$|j - i| \leq R;\ \forall\, j \in N\}$. Thus $N_{1,i} \subseteq N$ for all $i \in N$. Let σ_i denote the number of i's one-hop neighbors, i.e. $\sigma_i = |N_{1,i}|$. Now let $N_{m,i}$ denote i's m-hop neighbors, i.e. all nodes $j \in N_{m,i}$ can be reached with at most m hops from node i using the deployed routing protocol. Let ϱ_i denote the number of i's m-hop neighbors, i.e. $\varrho_i = |N_{m,i}|$. For an easier representation and without loss of generality, we denote i's one-hop neighbors as $j \in N_{1,i} = \{1, \ldots, \sigma_i\}$ and i's m-hop neighbors as $j \in N_{m,i} = \{1, \ldots, \varrho_i\}$, respectively, where i itself is part of $N_{1,i}$ and $N_{m,i}$.

5.2.4 Create Key Revocation Lists (KRLs)

Each node i creates a key revocation list $\mathcal{KRL}^i(t_x)$ containing all gathered accusation values for keys with the current expiry date t_x, i.e. accusations from its own neighborhood watch and received accusations from its neighbors. A revocation list $\mathcal{KRL}^i(t_x)$ can be represented as matrix as shown below

$$\mathcal{KRL}^i(t_x) = \begin{pmatrix} a^i_{1,1} & \cdots & a^i_{1,j} & \cdots & a^i_{1,\varrho_i} & ID_1 & v^i_1 & X^i_1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ a^i_{j,1} & \cdots & a^i_{j,j} & \cdots & a^i_{j,\varrho_i} & ID_j & v^i_j & X^i_j \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ a^i_{\varrho_i,1} & \cdots & a^i_{\varrho_i,j} & \cdots & a^i_{\varrho_i,\varrho_i} & ID_{\varrho_i} & v^i_{\varrho_i} & X^i_{\varrho_i} \end{pmatrix}, \tag{5.2}$$

in which accusation values are represented as $a^i_{k,j} \in \{0, 1, -\}$ with $\{k, j\} \in \{1, \ldots, \varrho_i\}$. Value $a^i_{k,j}$ indicates that node i "heard" that node j accuses node k of malicious behavior ($a^i_{k,j} = 1$) or, that j believes k is trustworthy ($a^i_{k,j} = 0$). Accusation value $a^i_{k,j} = -$ indicates that node k and node j are more than m hops apart, and thus are not allowed to give statements about each others trustworthiness. Accusation values $a^i_{i,i}$, i.e. $i = j = k$, indicate that node i revoked its own key, whereas accusation values $a^i_{j,i}$, i.e. $i = k$, indicate that node i accuses node j of malicious behavior. Both cases will be explained in Algorithms 2 and 1 in Section 5.3.1, respectively. All other accusation values are derived from received accusation messages which is explained in Algorithm 4 in Section 5.3.1. In the remainder of this chapter, we use $\mathcal{KRL}^i$ for short because all information in the revocation list are for public keys of

the current expiry date t_x. All expired public keys are automatically revoked and thus not listed any longer.

Each j-th column vector in $\mathcal{KRL}^i$ for $1 \leq j \leq \varrho_i$, short $\underline{c}_j^i$, contains all accusations $a_{k,j}^i$ made by node j against nodes $k \in N_{m,i}$. The upper index i denotes that values are current values in i's $\mathcal{KRL}^i$. Note that other nodes l might have different values stored in their revocation lists $\mathcal{KRL}^l$, e.g. $a_{k,j}^i \neq a_{k,j}^l$ for $i \neq l$ in some cases. Discrepancies in accusation values may exist, because accusations take time to propagate through the network. In addition, nodes have different m-hop neighborhoods and thus receive different accusation and harakiri messages.

Each j-th row vector in $\mathcal{KRL}^i$ for $1 \leq j \leq \varrho_i$, short $\underline{r}_j^i$, corresponds to a node $j \in N_{m,i}$ and contains, among other information, the accusation values $a_{j,k}^i$ from all nodes $k \in N_{m,i}$ evaluating node j. Hence, the i-th row in $\mathcal{KRL}^i$ contains all received accusations against node i itself. In particular, elements 1 to ϱ_i in $\underline{r}_j^i$ contain accusation values $a_{j,1}^i$ to a_{j,ϱ_i}^i. Element (ϱ_i+1) contains the identity ID_j of node j, the next element (ϱ_i+2) the current version number v_j^i of public key $Q_j(t_x)$. And the last element (ϱ_i+3) contains a 1-bit flag X_j^i that, when set, indicates that node i considers public key Q_j of node j as revoked. Node i sets

$$X_j^i = \begin{cases} 1 & \text{if} \quad a_{j,i}^i = 1 & \text{(Condition 1)} \\ & \text{or if} \quad a_{j,j}^i = 1 & \text{(Condition 2)} \\ & \text{or if} \quad \sum_k a_{j,k}^i \geq \delta \;\forall\; k \in N_{m,i} \text{ with } X_k^i = 0 & \text{(Condition 3)} \\ 0 & \text{else} \end{cases} \tag{5.3}$$

Basically, node i considers j's public key as revoked, i.e. $X_j^i = 1$, if at least one of Conditions 1-3 is true. Condition 1 describes the case that node i observed the malicious behavior of node j during its own neighborhood watch (see Revocation Algorithm 1 in Section 5.3.1). Condition 2 covers the case that i received a harakiri message from j indicating that private key d_j has been compromised (see Revocation Algorithm 2 in Section 5.3.1). And finally, Condition 3 defines that public key $Q_j(t_x, v_j^i)$ is revoked if node i received at least δ accusations against node j from trustworthy nodes k ($X_k^i = 0$) in i's m-hop neighborhood. Here, δ is a security parameter of our scheme. Note that "-" is treated as zero value in the summation. If

none of the three conditions applies, node i considers node j and its current public key $Q_j(t_x, v_j^i)$ as trustworthy, i.e. $X_j^i = 0$.

When first creating its key revocation list $\mathcal{KRL}^i$, node i initializes all accusation values with $a_{k,j}^i = 0$ for all $\{k, j\} \subset \{1, \ldots, \varrho_i\}$. As a consequence, all revocation values $X_j^i = 0$. This means we assume all nodes to be trustworthy until proven otherwise. In a more hostile environment, accusation values could be set to $a_{j,k}^i = 1$ until it has been observed that node j is indeed trustworthy. Note that accusation values $a_{k,j}^i$ cannot be initialized with "-" because node i does not know the hop distance between nodes j and k. This kind of information will be obtained as part of the $\mathcal{KRL}^i$ update, which we explain in Section 5.3.1. Once node i enters the network it starts its neighborhood watch and evaluates received accusations to update its $\mathcal{KRL}^i$. We assume fixed expiry intervals with $t_{x+1} = t_x + \Delta T$, i.e. all values in $\mathcal{KRL}(t_x)$ are re-set every ΔT.

5.2.5 Trust Model

First of all, we assume that the external KGC is honest, not compromised and trusted by all nodes. In applications where this is difficult to ensure, a distributed KGC can be deployed, e.g. [17, 19, 28, 51, 81, 99, 101]. Furthermore, as part of System Assumption 6, the KGC checks the identities of nodes before they receive their private keys to ensure that nodes only obtain keys corresponding to their identities.

The security and accuracy of our revocation scheme is based on the trust model defined in this section. We need the following definitions before we can derive the trust model:

Def. *Direct Accusations*: Accusations received from one-hop neighbors containing the result of their neighborhood watch.

For example, node i receives column vector $\underline{c}_j^j$ from an one-hop neighbor $j \in N_{1,i}$, where $\underline{c}_j^j$ contains the results of j's monitoring of all its one-hop neighbors $k \in N_{1,j}$.

Def. *Reported Accusations*: Accusations received from one-hop neighbors reporting accusations from nodes in l-hop distance, with $1 < l \leq m$.

For example, node i receives column vectors $\underline{c}_k^j$ from an one-hop neighbor $j \in$

$N_{1,i}$, where $k \in N_{m,i} \setminus N_{1,i}$.

Def. *Trusted Node*: Node i trust all nodes $j \in N_{m,i}$ with $X_j^i = 0$.

Note that we cannot call a node "trusted". The term has to be put in a relation, i.e. node i trusts node j, whereas another node k might not trust j. We are now ready to derive the trust model for our revocation scheme, in which each node i:

1. trusts that one-hop neighbors $j \in N_{1,i}$ that have been identified as malicious in i's neighborhood watch (with $a_{j,i}^i = 1$) are indeed malicious.
2. accepts direct accusations of any trusted one-hop neighbor $j \in N_{1,i}$.
3. accepts the majority vote of reported accusations from a group of at least ε trusted one-hop neighbors.
4. trusts δ or more accepted accusations (both direct and reported) against one node $j \in N_{m,i}$ to justify the revocation of j's keys.

The first item is based on the assumption that a monitoring scheme with very few false positives is used. For this reason, $a_{j,i}^i = 1$ leads to $X_j^i = 1$. Please note that a single trusted node cannot revoke a key. In fact, trust in a node follows that direct accusations are accepted and reported accusations by this nodes are considered for the majority computation. Node i counts all accepted accusations towards the δ revocation threshold. Only once at least δ accusations against the same node $j \in N_{m,i}$ have been accepted, j's keys are treated as revoked by node i, i.e. $X_j^i = 1$. This rule is reflected in Condition 3 in Eq. (5.3). Security parameter ε ensures that reported accusations from neighbors beyond i's own monitoring range have been observed by a group of trusted one-hop neighbors. This limits the impact of accusations that have been modified while propagating through the network. In our revocation scheme, δ and ε serve as security parameters that help preventing attacks by (colluding) malicious nodes and counteract inaccuracies of the implemented monitoring scheme as we demonstrate in our security analysis in Section 5.4.

There are some universal bounds for the security parameters, namely $1 \leq \varepsilon \leq \sigma_i$ and $1 \leq \delta \leq \varrho_i$. Typically $\delta \ll \varrho_i$ in order for our scheme to work, where the actual

value depends on the hostility of the network and network topology. We give tighter bounds for selecting these parameters such that attacks by colluding nodes are prevented in Section 5.4.

5.3 Key Revocation and Renewal for IBC Schemes

In this section, we propose a novel key revocation and key renewal scheme pairing-based for IBC schemes employed in MANETs.

5.3.1 Key Revocation

Every node in a MANET needs to be able to instantly verify whether a public key is revoked which requires that public key revocations are handled within the network in a self-organized fashion. Therefore we introduce a completely self-organized accusation-based revocation scheme with threshold δ that is based on monitoring one-hop neighbors. In our accusation-based scheme, the keys of node i are revoked either if node i notices that its own key has been compromised or if a node receives at least δ accusations against one node. We introduce four algorithms to provide key revocation in IBC schemes deployed in MANETs. In *Algorithm 1: Neighborhood watch*, nodes monitor the nodes in their neighborhood for suspicious behavior. *Algorithm 2: Harakiri* enables nodes to efficiently and securely revoke their own keys. In *Algorithm 3: Propagate*, accusations are securely sent to all neighbors. And finally in *Algorithm 4: Update KRL*, nodes update their key revocation lists using received accusations. Algorithms 1 and 2 require the propagation of messages to all neighbors, e.g. Algorithm 1 requires the propagation of observations and Algorithm 2 the propagation of so-called harakiri messages. Hence, Algorithm 3 is triggered by Algorithms 1 and 2, but also by Algorithm 4, namely whenever the key revocation list (KRL) of a node is updated as response to received accusation or harakiri messages. Basically Algorithms 3 and 4 create a loop that ensures that all accusations and harakiri messages are sent to all nodes in an m-hop neighborhood. An overview of the key revocation mechanism showing the interaction of the individual algorithms is depicted in Figure 5.1. The flow charts of the individual

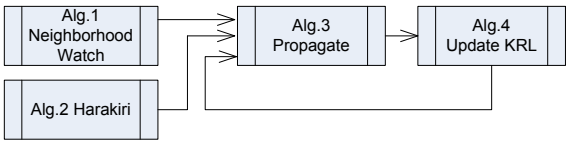


Figure 5.1: Overview of Key Revocation Scheme

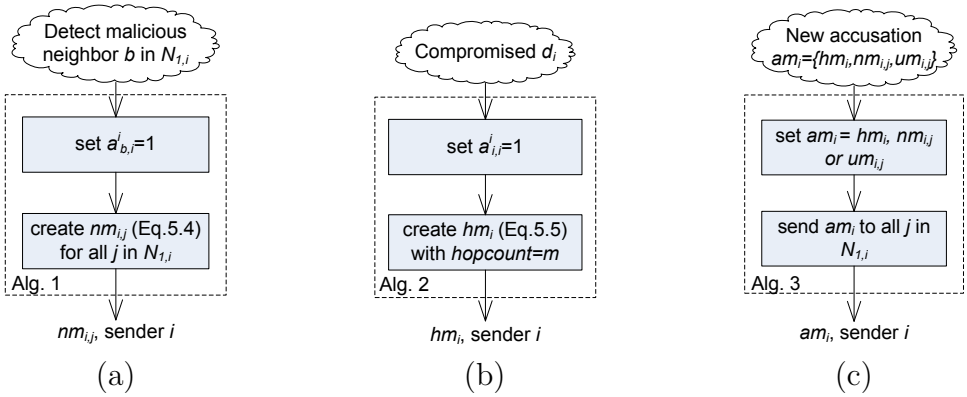


Figure 5.2: Flowchart: (a) Neighborhood Watch, (b) Harakiri, (c) Propagate.

algorithms are in Figures 5.2 and 5.3, in which the cloud symbol represents the event that triggered the algorithm, the dashed box contains all the steps of the algorithm, and the parameter at the end of the flowchart represent the output of the algorithm.

Algorithm 1: Neighborhood watch. The neighborhood watch scheme is a local monitoring scheme, in which each node i monitors all neighbors in its one-hop neighborhood $N_{i,1}$ for suspicious behavior. Metrics and tools for detecting suspicious behavior have been discussed in Section 5.1.2. The algorithm is depicted in Fig. 5.2-(a). As defined in Section 5.2.4, every network node i stores a key revocation list $\mathcal{KRL}^i$ as shown in Eq. (5.2). Please recall that the i-th column in $\mathcal{KRL}^i$, i.e. $\underline{c}_i^i$, corresponds to i's own accusations. Consequently, each time node i observes suspicious behavior of one of its own one-hop neighbors j, it updates the corresponding accusations values accordingly, i.e. $a_{j,i}^i = 1$. Node i can only monitor nodes j in its immediate communication range, i.e. $j \in N_{1,i}$, and thus only accusation values $a_{j,i}^i$ with $j \in N_{1,i}$ are updated during neighborhood watch.

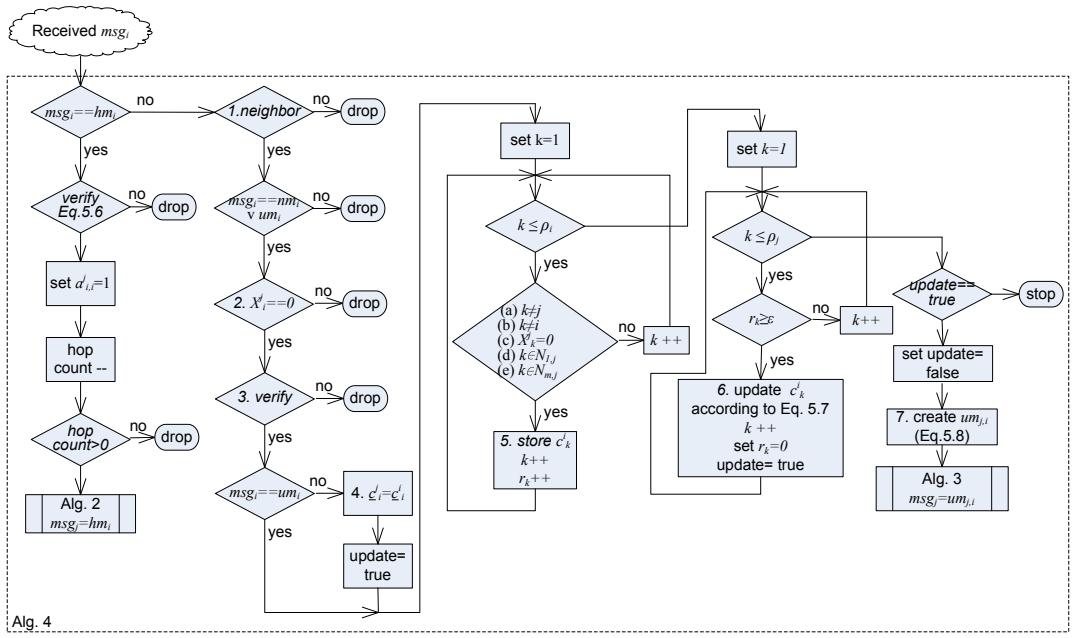


Figure 5.3: Flowchart Revocation Algorithm 4: Update KRL

Accusation values are updated every time i observes suspicious neighbors and at the beginning of every new expiry interval $t_{x+1} = t_x + \Delta T$. Once an accusation value $a^i_{j,i}$ is set, the value will not be reset to zero until a new public key $Q_j(v', t_x)$ with $v' > v^i_j$ is received or a new time interval t'_x starts.

Every time node i changes at least one accusation value $a^i_{j,i}$ from 0 to 1, i.e. from trustworthy to malicious status, i creates an neighborhood watch message $nm_{i,j}$ for each one-hop neighbor $j \in N_{1,i}$ according to Eq. (5.4) and starts *Algorithm 3* to propagate the accusations.

$$nm_{i,j} = (f_{K_{i,j}}(ID_i, nm_i), (ID_i, nm_i)), \text{ for all } j \in N_{1,i} \tag{5.4}$$

A neighborhood watch message $nm_{i,j}$ contains the identity of the sender, here ID_i, and the observations from i's neighborhood watch denoted as nm_i. For simplicity, we choose $nm_i = \mathcal{KRL}^i$, i.e. i submits its entire key revocation list. More bandwidth efficient solutions, e.g. only submitting values from i's neighborhood watch, i.e. $\underline{c}^i_i$,

or only updated values are discussed as extensions and improvements to the scheme. To avoid unauthorized or modified accusations, accusation messages are protected by a MAC function $f()$, where pairwise pre-shared keys from Eq. (4.4) serve as MAC keys. In particular, node i computes an accusation message $nm_{i,j}$ for each node $j \in N_{1,i}$ using $K_{i,j}$. The verification of received accusation messages is described in Algorithm 4. After computing all accusation messages, node i starts Algorithm 3 to propagate its observations.

Algorithm 2: Harakiri. The steps of the harakiri algorithm are illustrated in Fig. 5.2-(b). When a node i realizes that its private key d_i has been compromised, i creates a harakiri message hm_i as shown in Eq. (5.5) below.

$$hm_i = (ID_i, d_i, Q_i, (t_x, v_i), \text{“revoke”}, hopcount) \tag{5.5}$$

The message contains the sender's identity (ID_i), the compromised private key d_i, the corresponding public key Q_i, the expiry date and version number of the public key (t_x, v_i), and a text string that marks the message as revocation message. The last field in the harakiri message is the *hopcount* that ensures that the message reaches all nodes in m-hop distance to i, i.e. all nodes $j \in N_{m,i}$. Therefore, sender i initially sets $hopcount = m$. Note that only node i and the entity that compromised i are in possession of d_i and are thus the only entities that can create a valid harakiri message hm_i. Since the adversary has no motivation to revoke the key, the harakiri message does not need to be authenticated or otherwise protected. The verification process of received harakiri messages checks wether d_i is a valid private key and corresponds to Q_i and ID_i and is discussed in Algorithm 4. Every time a node i detects its compromise and creates a harakiri message hm_i, i will start Algorithm 3 to propagate the message.

Algorithm 3: Propagate. In this algorithm nodes securely propagate accusations to their one-hop neighbors. The steps are illustrated in Fig. 5.2-(c). Accusation messages am_i can be neighborhood watch messages $nm_{i,j}$ (see Eq. (5.4)), harakiri messages hm_i of a compromised node i (see Eq. (5.5)), or key revocation update messages $um_{i,j}$ from Algorithm 4 (see Eq. (5.8)), i.e. $am_i \in \{nm_{i,j}, hm_i, um_{i,j}\}$. As

illustrated in Figure 5.1, Algorithm 3 is triggered by Algorithms 1, 2 and 4. The initiator of Algorithm 3, i.e. sender i of am_i, sends its accusation message(s) to all its one-hop neighbors $j \in N_{1,i}$. Note that for $am_i = hm_i$ there is only one message that is broadcasted to all neighbors, whereas for $am_i = nm_{i,j}$ or $am_i = um_{i,j}$ there are σ_i messages, i.e. one for each neighbor j, that are unicasted to each neighbor.

Algorithm 4: Update KRL. In this algorithm, node i updates its key revocation list $\mathcal{KRL}^i$ using the received accusation messages am_j from its neighbors j. Thus, Algorithm 4 is triggered by Algorithm 3 (see Figure 5.1), and key revocation lists are updated every time a new accusation message is received. We distinguish between three types of updates according to the received message am_j and we describe the update process for received harakiri, neighborhood watch, and update messages separately in the following paragraphs. The algorithm is the most complex one in the revocation scheme and requires several processing steps, as illustrated in the flow chart in Figure 5.3.

Received $am_j = hm_j$. The receiver i of a harakiri message hm_j needs to verify whether the message is authentic. As mentioned earlier, it is not necessary to check who sent the message (which is why hm_j does not provide not message authentication), because the public key that corresponds to the broadcasted private key d_j should not be used any longer regardless of the sender of the message. However, it needs to be verified if the broadcasted private key d_j corresponds to public key Q_j and identity ID_j. Only then, public key Q_j should be revoked. Otherwise, adversaries could fabricate false harakiri messages that cause public keys of uncompromised nodes j to be revoked. Node i verifies whether hm_j is valid by checking wether Eq. (5.6) below is true.

$$K_{i,j} = \hat{e}(d_j, Q_i) \tag{5.6}$$

The check verifies whether the broadcasted private key d_j indeed corresponds to the public key Q_j and thus ID_j. Therefore, a recipient of hm_j, here node i, looks up whether it is in possession of public key Q_j and the pre-shared key $K_{i,j}$ and if so, uses the $K_{i,j}$ to check whether Eq. (5.6) is true. If i is not in possession of these

keys, i first computes Q_j from the received ID_j, t_j and v_j according to Eq. (5.1) and checks whether the received ID_j and the computed Q_j correspond to each other. If this check is successful, i derives $K_{i,j}$ according to Eq. (4.4). Finally, i checks whether Eq. (5.6) is true. If the check is successful, i updates its key revocation list $\mathcal{KRL}^i$ by setting accusation value $a^i_{j,j} = 1$. Hence, Condition 1 in Eq. 5.3 is satisfied and i sets $X^i_j = 1$ and thus considers $Q_j(t_x, v_j)$ revoked. Then i decrements the *hopcount*, i.e. $hopcount := houpcount - 1$ and if $hopcount > 0$, node i starts Algorithm 3 with $am_i = hm_j$. If one of the check fails, i discards hm_j and aborts the algorithm.

Received $am_j = nm_{j,i}$. When node i receives a neighborhood watch message $nm_{j,i}$ from one of its neighbors $j \in N_{1,i}$, i first needs to verify if the message is authentic. If the check is successful, i uses the received message nm_j to update its key revocation list $\mathcal{KRL}^i$. In the following we describe all necessary steps of the verification and update process. Note that, if a step is successful, i continues with the next step, else i drops the packet and aborts the algorithm. For efficiency reasons, node i first checks if the sender of the message is a trusted one-hop neighbor before executing the (potentially) computationally more demanding message authentication. Upon receiving $am_j = um_{j,i}$, node i performs the following steps:

1. *neighbor check*: i checks whether sender j is a direct neighbor, i.e. $j \in N_{1,i}$.
2. *check trustworthiness*: i checks whether j is considered trustworthy, i.e. $X^i_j = 0$.
3. *verify message authenticity*: i verifies the MAC of the received message $nm_{j,i}$ using pre-shared key $K_{i,j}$. If i is currently not in possession of $K_{i,j}$, i first computes the key according to Eq. (4.4).
4. *copy neighbor's observation*: i extracts column vector $\underline{c}^j_j$ from nm_j to update its own column vector $\underline{c}^i_j$ in $\mathcal{KRL}^i$. In other words, i adopts all accusation values from j's neighborhood watch. Here, node i copies only accusation values of nodes that are in i's own m-hop neighborhood, i.e. $a^j_{k,j}$ for all $k \in N_{m,i}$. All other accusation values are discarded. Upon completion, node i sets the update flag, i.e. $update = true$.

5. *store other nodes' observations*: i scans through all columns $\underline{c}_k^j$ with $k \in \{1, \ldots, \varrho_j\}$ in $\mathcal{KRL}^j$ and stores all columns $\underline{c}_k^j$ for which *all* following conditions hold:

   (a) $k \neq i$

   (b) $k \neq j$

   (c) $X_k^i = 0$

   (d) $k \notin N_{1,i}$

   (e) $k \in N_{m,i}$

   Columns $\underline{c}_k^j$ that do not satisfy Condition (a) are discarded because the column corresponds to i's own neighborhood watch. Condition (b) is necessary because j's neighborhood observations have already been adopted in Step 4. Condition (c) ensures that only accusations from trustworthy nodes k are used. Condition (d) discards j's copy of observations of i's other one-hop neighbors, because i receives these observations directly from these one-hop neighbors. The last condition ensures that i only stores accusations from nodes in its own m-hop neighborhood which might be different from j's m-hop neighborhood.

   Node i checks Conditions (a)-(e) for all $k \in \{1, \ldots, \varrho_j\}$. If all conditions are met for k, i stores $\underline{c}_k^j$ and increments counter r_k. All other columns are discarded. We refer to the stored vectors as k-vectors. These vectors are not directly used to update key revocation list $\mathcal{KRL}^i$ because the vectors contain "second" or even worse n-th hand information. As discussed in our trust model in Section 5.2.5 we need a minimum of ε received k-vectors to establish trust in the reported accusations of node k. If at least ε k-vectors, i.e. $r_k \geq \varepsilon$ are collected, i updates its $\mathcal{KRL}^i$ as described in the next step.

6. *use accumulated k-vectors for update*: Node i checks for all $k \in \{1, \ldots, \varrho_i\}$ whether $r_k \geq \varepsilon$. If true, i.e. i stored k-vectors $\underline{c}_k^j$ from at least ε one-hop neighbors j, node i updates the k-vector in $\mathcal{KRL}^i$ as described in the following.

For an easier representation and without loss of generality, we assume i stored r_k column vectors $\underline{c}_k^j$ from r_k one-hop neighbors j with $j \in \{1, \ldots, r_k\}$ in Step 5, with $r_k \geq \varepsilon$. Each accusation value $a_{l,k}^i$ with $l \in \{1, \ldots, \varrho_i\}$ in $\underline{c}_k^i$ is computed from the majority vote over all collected $a_{l,k}^j$, with $j \in \{1, \ldots, r_k\}$ as shown in Eq. (5.7).

$$a_{l,k}^i = \begin{cases} 1 & \text{if } \sum_{j=1}^{r_k} a_{l,k}^j > \frac{r_k}{2} \\ 0 & \text{if } \sum_{j=1}^{r_k} a_{l,k}^j < \frac{r_k}{2} \\ a_{l,k}^i & \text{else} \end{cases} \tag{5.7}$$

Basically, if more than halve of the accumulated accusation values $a_{l,k}^j$ against a node l equal 1, node i sets the value to 1. If more than halve of the accumulated values are 0, i sets the value to 0. If no majority can be found, the accusation value in $\mathcal{KRL}^i$ remains unchanged. Note that the range for index l of the accumulated accusation values $a_{l,k}^j$ is $l \in \{1, \ldots, \varrho_j\}$, whereas in the KRL update, node i only considers values of nodes l in its own m-hop neighborhood, i.e. $l \in \{1, \ldots, \varrho_i\}$, where ϱ_j might be different from ϱ_j. Accusation values $a_{l,k}^j$ of the accumulated k-vectors with $l \notin N_{m,i}$ are discarded in the update calculations. Note that all k-vectors that have been used for the KRL update are erased, whereas "unused" k-vectors with $r_k < \varepsilon$ remain in i's storage. If $r_k \geq \varepsilon$ for at least one k, i.e. i updated at least one k-vector in $\mathcal{KRL}^i$, node i sets the update flag $update = true$.

7. *prepare update message*: If $update = true$, which is always true for $am_i = nm_{i,j}$, node i prepares an update message $um_{i,j}$ for all its one-hop neighbors $j \in N_{1,i}$ according to Eq. (5.8). The messages are constructed similar to the neighborhood watch messages $nm_{i,j}$ in Eq. (5.4), where the difference is not in the message but rather in the treatment of received accusation messages $am_i = nm_{i,j}$ or $am_i = um_{i,j}$. For simplicity, we assume $um_i = \mathcal{KRL}^i$, where more bandwidth efficient solutions such as only sending updated vectors are possible. Each message is protected with pre-shared key $K_{i,j}$ serving as MAC key. All messages $um_{i,j}$ for all $j \in N_{1,i}$ serve as input to Algorithm 3 and thus

are propagated to i's one-hop neighborhood. After triggering Algorithm 3, the update flag is reset, i.e. $update = false$.

$$um_{i,j} = (f_{K_{i,j}}(ID_i, um_i), (ID_i, um_i)), \text{ for all } j \in N_{1,i} \tag{5.8}$$

Received $am_j = um_{j,i}$. If node i receives a KRL update message from j, i proceeds almost as in the previously described case for received neighborhood watch messages nm_i. More precisely, upon receiving $am_i = um_{i,j}$, node i executes Steps 1-3 and 5-7 as described in the previous paragraph. In other words, node i updates its key revocation list $\mathcal{KRL}^i$ identical to the previously described case, except that i does not copy accusations from j's neighborhood watch (Step 4). If node i updated at least one of its k-vectors, i.e. $update = true$, i creates an update messages $um_{i,j}$ according to Eq. (5.8)for all $j \in N_{1,i}$ and starts Algorithm 3 as described in Step 7.

5.3.2 Example for $\mathcal{KRL}$ Update

We present an artificially small and simple network scenario to illustrate how Algorithm 4 works for a received neighborhood watch message, i.e. for $am_i = nm_{i,j}$. Note that this example also covers $am_i = um_{i,j}$, because both scenarios only differ in Step 4. In our example we consider six network nodes i with $i \in \{1, \ldots, 6\}$ with a network topology as shown in Figure 5.4. In the figure all nodes that are in each others direct communication range are connected by a solid line, which corresponds to one hop. The nodes maintain key revocation lists for nodes in two hop communication range, i.e. $m = 2$, and the security parameters are set to $\delta = 3$ and $\varepsilon = 2$. We can observe from the figure node 1's one-hop neighborhood $N_{1,1} = \{1, 2, 3, 4\}$ and its two-hop neighborhood $N_{2,1} = \{1, 2, 3, 4, 5\}$. We now show how node 1 updates its revocation list $\mathcal{KRL}^1$ upon receiving accusation messages $am_2 = nm_{2,1}, am_3 = nm_{3,1}, am_4 = nm_{4,1}$ from its one-hop neighbors 2, 3 and 4, respectively. To update its key revocation list, node 1 executes Algorithm 4 for each received accusation message. For simplicity, we assume that all public keys Q_2, Q_3 and Q_4 are not expired and have a version number $v = 1$. Hence, we neglect values (t_x, v) in our example. Node 1's revocation list from before the update as

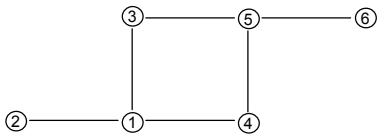


Figure 5.4: Network Topology in Toy Example

well as the extracted key revocation lists from am_3 and am_4 are shown below.

$$\mathcal{KRL}^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & ID_1 & 0 \\ 1 & 0 & 0 & 0 & - & ID_2 & 1 \\ 0 & 0 & 0 & 0 & 1 & ID_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & ID_4 & 0 \\ 0 & 0 & 0 & 0 & 0 & ID_5 & 0 \end{pmatrix},$$

$$\mathcal{KRL}^3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & - & ID_1 & 0 \\ 1 & 0 & 1 & 1 & - & - & ID_2 & 1 \\ 0 & 1 & 0 & 0 & 0 & - & ID_3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & ID_4 & 0 \\ 0 & - & 1 & 1 & 0 & 1 & ID_5 & 1 \\ - & - & 1 & 1 & 0 & 0 & ID_6 & 1 \end{pmatrix}, \quad \mathcal{KRL}^4 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & - & ID_1 & 0 \\ 1 & 0 & 1 & 1 & - & - & ID_2 & 1 \\ 0 & 1 & 0 & 0 & 1 & - & ID_3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & ID_4 & 0 \\ 0 & - & 1 & 1 & 0 & 1 & ID_5 & 1 \\ - & - & 1 & 1 & 1 & 0 & ID_6 & 1 \end{pmatrix}.$$

For brevity of the presentation, we consider a parallel execution of Algorithm 4 for all received accusation messages and we discuss the steps in the following:

1. Nodes 2, 3 and 4 are all one-hop neighbors (see Figure 5.4).

2. Nodes 3 and 4 are considered trustworthy because $X_3^1 = X_4^1 = 0$ in $\mathcal{KRL}^1$, whereas node 2 is marked as malicious ($X_2^1 = 1$) and thus am_2 is discarded.

3. Node 1 successfully authenticates am_3 and am_4 using $K_{1,3}$ and $K_{1,4}$, respectively.

4. Node 1 uses column vectors $\underline{c}_3^3$ from $\mathcal{KRL}^3$ and column vector $\underline{c}_4^4$ from $\mathcal{KRL}^4$ to update column vector $\underline{c}_3^1$ and $\underline{c}_4^1$ in $\mathcal{KRL}^1$, respectively. The updated vectors in $\mathcal{KRL}^i$ are:

$$\underline{c}_3^1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \underline{c}_4^1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

Then node 1 sets $update = true$.

5. Node 1 scans through all columns in $\mathcal{KRL}^3$ and discards the following columns: $\underline{c}_1^3$ because this column contains 1's own reported accusations, $\underline{c}_2^3$ because node 1 does not trust node 2 ($X_2^1 = 1$), $\underline{c}_3^3$ because $3 \in N_{1,1}$ and 1 used 3's direct accusations already in Step 4, $\underline{c}_4^3$ because $4 \in N_{1,1}$ and 1 trusts 4's direct accusations more than 3's reported accusations, and $\underline{c}_6^3$ because $6 \notin N_{2,1}$. Hence, node 1 only stores $\underline{c}_5^3$ from $\mathcal{KRL}^3$. For similar arguments, node 1 stores only $\underline{c}_5^4$ from $\mathcal{KRL}^4$.

6. Node 1 checks for all $k \in \{1, 2, \ldots, 5\}$ whether $r_k \geq \varepsilon$, which has only one element $k = 5$ with $r_5 = 2 \geq \varepsilon = 2$. Consequently, node 1 uses those two vectors to update $\underline{c}_5^1$ in $\mathcal{KRL}^1$. Using the majority vote over $\underline{c}_5^3$ and $\underline{c}_5^4$ from Eq. (5.7), node 1 obtains its new vector $\underline{c}_5^1$ as illustrated below. Node 1 sets $update = true$.

$$\underline{c}_5^3 = \begin{pmatrix} 1 \\ - \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \underline{c}_5^4 = \begin{pmatrix} 0 \\ - \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad \underline{c}_5^1 = \begin{pmatrix} 0 \\ - \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

Combining all updates, node 1 obtains a new $\mathcal{KRL}^1$ as shown below.

$$\mathcal{KRL}^1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & ID_1 & 0 \\ 1 & 0 & 1 & 1 & - & ID_2 & 1 \\ 0 & 0 & 0 & 0 & 1 & ID_3 & 0 \\ 0 & 0 & 0 & 0 & 1 & ID_4 & 0 \\ 0 & 0 & 1 & 1 & 0 & ID_5 & 0 \end{pmatrix}$$

7. Since $update = true$, node 1 prepares an update message $um_{1,j}$ for $j = \{2, 3, 4, 5\}$ according to Eq. (5.8). Finally, node 1 starts Algorithm 3 to propagate these messages to its one-hop neighbors.

5.3.3 Key Renewal

The presented IBC revocation scheme for MANETs needs to be complemented by a key renewal algorithm to enable node i to obtain a new key pair (Q_i, d_i) after its public key expired, or was revoked by a harakiri message or δ accusation messages. In any case, a node needs to access the off-line KGC for key renewal. When doing so, the node must re-authenticate itself to the KGC using some credentials that identify the node. An off-line KGC cannot distinguish between malicious nodes whose keys have been revoked because of bad behavior or honest nodes whose keys have been compromised. Therefore, malicious nodes can always request new keys once their old keys have been revoked due to malicious behavior. However, these malicious nodes must act under their true identities in order to successfully authenticate themselves to the KGC. To restrict the power of such malicious nodes, we select a maximum version number v_{max}, i.e. the number of key renewals for the same expiry date is restricted. Clearly, a node that requests more than v_{max} key renewals in one expiry date interval is either malicious or not able to appropriately protect its key data.

If node i received a new key Q_i' with version number $v_i' > 1$, i needs to broadcast its new public key to its m-hop neighborhood after re-joining the network. The receivers of Q_i', update the version number in their revocation lists accordingly and set all accusation values for Q_i' to zero. If only the expiry date is new and $v_i' = 1$, i does not need to inform other nodes about its new keys. Every node in $N_{m,i}$ automatically updates all accusation values and revocation flags in its key revocation lists at every new expiry interval, i.e. when $t > t_x$ and the new expiry date of all keys is set to $t_{x+1} = t_x + \Delta T$. Consequently, the key revocation lists are re-set every ΔT.

5.3.4 Extensions

Many extensions to the proposed key revocation and key renewal schemes are possible. We briefly outline some of them in this section.

- *Distributed On-line KGC.* Our schemes can be easily modified for MANETs with distributed on-line KGCs, where the revocation scheme remains unchanged and the distributed on-line KGC takes over the task of key renewal. In that way external KGCs are only used for the initial key distribution (availability scenario AV-3) or not at all (AV-4). Note that latter case removes System Assumption 6. For example, (k, n)-threshold schemes can be used to distribute master key s to all network nodes such that k D-KGCs can collaboratively generate and distribute new private keys during key renewal, as in [33, 77, 85, 124, 125]. Scenario B from the *Distribute* Algorithm in the basic framework (see Section 4.3.2) is desirable for key renewal, because it does not require confidential channels between node and on-line KGCs. Since our revocation scheme works completely independent of the (k, n)-threshold schemes, the solution is very efficient. In our scheme with distributed KGC, all nodes serve as D-KGC and only one master key s is used. Hence, unlike in [124], the compromise of k nodes compromises the entire system as it does in all other schemes using a single master key that is distributed using a (k, n)-threshold scheme, e.g. [33, 77, 85, 125].

- *Weighted Accusations.* In another possible extension to our presented revocation scheme, weighted accusation values are used, as introduced in [32]. Hence, instead of using binary values $\{0, 1\}$ to represent accusation values $a^i_{j,l}$, accusation values are real numbers in interval 0 and 1, i.e. $[0, 1]$. The value can be based on several parameters such as number of accusations a node has made, number of accusations against a node or a certainty value generated by the monitoring scheme.

- *Confidential Accusations.* To avoid that malicious nodes can overhear accusations against them and use this knowledge to keep the number of their accusations below δ, accusations can be encrypted, as suggested in [124]. Unlike

stated in [124], we do not think that malicious nodes that stop misbehaving after noticing a number of accusations ($< \delta$) against themselves pose a threat. We rather believe that it does not matter whether nodes are just pretending to be trustworthy or they are indeed trustworthy, because as long as the nodes behave well they do not launch an attack. For instance, a previously selfish node might realize that it is time to start forwarding messages again. However, we believe that nodes may use the knowledge about their own accusations to move to a new m-neighborhood whenever their accusation count approaches δ. We refer to these kind of malicious nodes as *roaming adversaries*. For accusation confidentiality, any symmetric encryption algorithm can be used with a symmetric encryption key $K'_{i,j}$ that is derived from the pre-shared keys $K_{i,j}$ in Eq. (4.4). In that case, all neighborhood watch and update messages are encrypted and only sent to neighbors that are not accused in these messages, e.g. $nm_{i,j} = (f_{K_{i,j}}(ID_i, nm_i), E_{K_{i,j}}(ID_i, nm_i))$, for all $j \in N_{1,i} \setminus \{L\}$, where L is the set consisting of all neighbors that have not been accused by either any node, node i's one-hop neighbors or just node i, depending on the implementation. We conclude that encrypting accusations helps to thwart roaming adversaries while sacrificing the motivational factor of overheard accusations.

- *Dedicated Key Pairs.* Harakiri messages hm_i (see Eq. (5.5)) contain private keys d_i which affects the security of all previous messages that were either signed under d_i or encrypted under Q_i. For example, an adversary who receives hm_i can decrypt all messages encrypted under Q_i as well as all messages encrypted under $K_{i,j}$ for any $j \in N$. In addition, adversaries can forge signatures that look like they have been created before the signing key was revoked. To prevent the described misuses of self-revoked keys, we suggest using dedicated private and public key pairs for different purposes. For example, public keys could contain a label that specifies the purpose of the keys, i.e.

$$Q_i(t_x, v_i) = H_1(ID_i||t_x||v_i||\text{label}),$$

where label $\in$ {sign, encrypt, revocation}. This format still allows nodes i

to derive public keys Q_j and pre-shared keys $K_{i,j}$ of other nodes j in a non-interactive fashion. When using this key format, private keys used in the revocation scheme can be revealed in harakiri messages without revealing private keys dedicated to other purposes. In addition, once a public key used in the revocation scheme is revoked, all other public keys of the same node are revoked as well.

- *Adapting Scheme to Hostile Environments.* The presented key revocation scheme can be adapted to many different environments, e.g. more hostile ones. For instance, a different majority function for computing accusations than the one in Eq. (5.7) can be defined, accusation values could be initialized with ones instead of zeros and different triggers for when accusation messages are sent can be defined. These parameters should be selected according to the fraction of expected malicious nodes, i.e. the hostility of the MANET environment.

- *Adaptive Monitoring Schemes and Security Parameters.* As mentioned in the previous paragraph, monitoring scheme and security parameters should be selected according to the hostility of the implementation environment. However, sometimes it might be advisable to adjust these values in the running system. For instance, some nodes might have a number of accusations always just below threshold δ. In that case the threshold of these or all nodes should be dropped accordingly at the next expiry interval. If nodes still receive the same number of accusations, their keys will be revoked during the new expiry interval. In other cases it might be advisable to have an adaptive monitoring scheme. For instance, if a node is at the edge of the network it does not need to forward many packets. In that case, a monitoring scheme that only monitors the number of forwarded packets is not able to evaluate the maliciousness of these nodes. Here it would be advisable to have an adaptive monitoring scheme that implements several kinds of metrics and thresholds that can be selected according to some network parameters, such as network density, number of neighbors, position in the network, etc..

- *Efficiency Improvements.* The efficiency of the proposed basic schemes can be significantly increased by only sending new accusation as part of the neighborhood and update messages, as opposed to entire key revocation lists. Another way to improve the efficiency of the revocation scheme is based on the selection of when accusation messages are sent. For example, instead of sending update messages each time one accusation value changes, messages could only be sent when the revocation status X_i of a key changes, i.e. $X_i = 0 \rightarrow X_i = 1$, or until a node accumulated at least γ accusations, or at fixed time intervals τ. However, note that the frequency updates are sent constitutes a security-performance trade-off, because less frequent updates increase the performance while the security might be reduced due to longer propagation delays of accusations. In general, the frequency should be chosen according to the hostility of the network and could be implemented in an adaptive manner. Furthermore, rules can be implemented to define priority levels for different message types, e.g. harakiri messages could have highest priority and be sent out instantly, changes in revocation status have lower priority and are sent out in the next time interval τ, whereas other changes of accusation values are collected until a certain threshold γ is reached.

- *Network-wide Revocations.* The propagation range m of accusations can be removed in the revocation scheme. In that case accusations are sent to all network nodes and thus nodes store information for all network nodes and their public keys. The disadvantage of this modification is the increased communication load for the entire network and the increased storage requirements for larger key revocation lists. Hence, m serves as performance parameter that can be selected according to the number of malicious nodes, available network bandwidth and power constraints of nodes. Removing m completely enables the implementation of the revocation scheme in networks in which nodes do not know which nodes are in their m-hop neighborhood (i.e. system assumption 6 in Section 5.2.1 is eliminated).

- *Adversary Models.* In the described revocation scheme, the implemented mon-

itoring scheme is treated as a black box with false positive and false negative rates α and β, respectively. In other words, we assume that nodes mark their one-hop neighbors as malicious or trustworthy with accuracy determined by α and β, but we specify neither how malicious behavior is defined nor how such behavior can be measured. Some possible metrics and detection mechanisms are summarized in Section 5.1.2. In order to thwart particular attacks and/or specific malicious behavior, the monitoring scheme must be modelled according to the considered adversary model. In other words, the monitoring scheme is set to detect the signature of a certain attack modelled in the adversary model. For instance, if the adversary model describes adversaries launching DoS attacks, the monitoring scheme could be set to detect large numbers of sent packets. Or if the adversary model describes adversaries who launch specific routing attacks such as blackhole attacks, the monitoring scheme could be set to check the number of dropped packets. Since the monitoring scheme is treated as black box, the presented revocation scheme can be used to thwart all types of existing as well as potential future attacks. Only the monitoring scheme must be adapted to the respective adversary model while the revocation scheme can remain unchanged.

5.4 Security Analysis

We assume the underlying IBC scheme including the pre-shared keys from Eq. (4.4) to be secure and refer to Section 4.5.1 for a security discussion of the ID-based framework. Henceforth, we limit our analysis to the introduced key revocation and key renewal schemes. In Section 5.4.1 we show that our proposed key renewal scheme resists Sybil and impersonation attacks. In Section 5.4.2, we show that our proposed key revocation scheme prevents attacks by outsiders, and in Sections 5.4.3-5.4.6 we analyze the scheme's resilience to non-colluding malicious nodes, falsely accused nodes, colluding one-hop neighbors, and colluding l-hop neighbors, respectively.

5.4.1 Sybil and Impersonation Attacks on Key Renewal Scheme

Malicious nodes could try to bypass security parameter δ by fabricating δ different identities in a so-called Sybil attack [35]. In that scenario, a single node can send δ accusations against node i and thus revoke i's key. Our scheme uses ID-based public keys of a fixed format, i.e. upon identifying to the KGC, a node can only obtain one possible valid private key. Hence, Sybil attacks are prevented in our scheme, because of the use of ID-based keys and the fact that the off-line KGC checks the identity of every node before issuing keys.

Adversaries who have impersonated network nodes, cannot request new keys upon their old keys have been expired or revoked, because the impersonators cannot successfully authenticate to the KGC. On the other hand, malicious nodes that act maliciously under their own identity are able to request new keys, however the number of renewals in one time interval ΔT is limited to v_{max}. In addition, the time span for attacks by malicious nodes is limited to the time period between key renewal and subsequent key revocation in the neighborhood watch scheme.

We can conclude that the security of the key renewal scheme is based on the honesty of the KGC and the procedure that is used to verify nodes' identities.

5.4.2 Outsider Attacks on Revocation Scheme

In the revocation scheme, all neighborhood watch and update messages are protected with pre-shared keys $K_{i,j}$. Thus the messages provide message authentication and integrity protection and can neither be fabricated nor modified by outsiders of the network. On the other hand, harakiri messages are not protected but contain private and public key pairs. Hence, the messages can only be created by insiders or adversaries who compromised a network node. Note that latter have no reason to send harakiri messages, because the message would cause the revocation of the compromised keys and thus render its compromise useless. If dedicated key pairs are used, as introduced as possible extension, the self-revocation of keys does not affect any previously signed or encrypted messages of the same node.

An outsider could attempt draining a node's battery in a so-called battery exhaustion attack [115] by repeatedly sending messages that cause a node to perform demanding computational operations. These attacks are prevented by our revocation scheme, because nodes only accept accusation messages from trusted one-hop neighbors. This check is very efficient (basically it is just a look up of the sender's identity and revocation status in the key revocation list) and thus cannot be exploited to drain the battery. If an adversary spoofs the identity of a trusted one-hop neighbor, the attack would be detected when verifying the authenticity of the first message. Even though the verifying process is more demanding, the process cannot be repeated because the spoofed identity is marked as malicious after the first message failed to authenticate successfully. Consequently, no more messages originating from the same source will be accepted, preventing the attack.

We conclude that outsiders to the network cannot attack the revocation scheme.

5.4.3 Selfish, Malicious, and Roaming Adversaries

We believe that node compromises are likely to occur in MANETs due to weak physical protection of nodes and potentially hostile network environments. In addition, selfish nodes may exist in some applications. If the metrics of the implemented monitoring scheme are selected accordingly, compromised and selfish nodes can both be detected in our neighborhood watch scheme. In our revocation scheme, keys from malicious nodes are first locally revoked by one-hop neighbors who witnessed malicious behavior of these nodes as part of their neighborhood watch. These witnesses then propagate accusations, which, once enough accusations have been accumulated, lead to key revocations by all nodes in m-hop distance. Note that keys are never globally revoked, each node i rather has their individual view on which keys it considers as revoked based on its accusation values in $\mathcal{KRL}^i$. Hence, our revocation scheme excludes adversaries who control compromised nodes from the network, because their keys will be revoked and they cannot request new keys. In addition, selfish nodes that do not participate in distributed network tasks such as forwarding messages, will have their keys revoked. This forces selfish nodes to frequently renew their keys, which imposes higher costs than performing the ini-

tially requested network tasks. Hence, our revocation scheme encourages selfish nodes to participate in network tasks.

A single malicious node can only send one accusation for each network node, i.e. a single node cannot revoke a key of another node. In particular, each node k may send more than one accusation against the same node j, however each receiver i of these accusation message only stores one accusation value $a^i_{j,k}$, namely the most recent one.

An undetected malicious node attempting to launch a battery exhaustion attack (as described in the previous section), could send messages that would be initially accepted because they pass the verifications. However, eventually the attacker would be marked as malicious as part of the neighborhood watch. The monitoring scheme can be set such that the attack is detected before the battery of a node is exhausted.

Malicious nodes cannot simply drop accusations against themselves, because this will be detected by the neighborhood watch scheme. Besides, accusations are broadcasted by all neighbors of a node and thus still reach other nodes, even if one of the propagation paths is broken. Attempts of malicious nodes to modify accusations against themselves are prevented by using integrity protected accusations. However, an adversary can modify its own key revocation list before sending it to all neighbors. The impact of this attack is limited by security parameters δ and ε and will be discussed for colluding adversaries in the next sections.

A roaming adversary i may move to a new neighborhood every time its number of accusation approaches δ. However, i's new one-hop neighbors will eventually detect i's malicious behavior and thus i needs to move again before its key is revoked. Lets assume nodes are uniformly distributed and each routing hop is over a distance R. Then, the speed S that is necessary for roaming adversaries to travel to a new m-hop neighborhood before their current keys expire at time t_x is

$$S \geq \frac{mR}{t_x - t}.$$

Note that t is the current time and thus $t_x - t \leq \Delta T$. We can observe that by

selecting m sufficiently large and the expiry intervals ΔT small, an adversary has to travel very fast to escape the revocation of its key. Hence, the capabilities of roaming adversaries are fairly limited, because adversaries need to move fast and cannot remain at the same location for a longer period of time. In addition, if accusations are encrypted as described as one possible extension in Section 5.3.4, nodes cannot learn about the number of accusations against them. Thus, nodes do not know when they should move to a new neighborhood, which further limits the power of roaming adversaries.

5.4.4 Falsely Accused Nodes

The introduced revocation scheme relies on monitoring one-hop neighbors and propagating the observations. Hence, the scheme's security and accuracy depend on its resilience to colluding malicious nodes, as well as the false positive rate α and false negative rate β of the employed monitoring scheme. We assume that all nodes implement the same monitoring scheme.

For our analysis we need to introduce some notations, summarized in Table 5.1. Lets H_i denote i's honest one-hop neighbors, with $|H_i| = n_h^i$, and M_i i's malicious one-hop neighbors, with $|M_i| = n_m^i$. That follows that $N_{1,i} = H_i \bigcup M_i$, where $H_i \bigcap M_i = \emptyset$ and thus $\sigma_i = n_h^i + n_m^i$. Furthermore, F_i denotes i's honest one-hop neighbors that have been falsely marked as malicious by i, with $|F_i| = n_f^i$, and U_i denotes i's undetected malicious one-hop neighbors, with $|U_i| = n_u^i$. Hence, $F_i \subseteq H_i$ and $U_i \subseteq M_i$. Finally, the colluding nodes are denoted as C with $|C| = n_c$. In our analysis of colluding one-hop neighbors, we consider the case that all undetected malicious nodes collude, i.e. $C = U_i$ and $n_c = n_u^i$.

We can observe that false positive rate α causes a node to falsely mark $n_f^i = \alpha n_h^i$ of its honest one-hop neighbors as malicious. Consequently, from item 1 in our trust model in Section 5.2.5, a node will revoke the keys of n_f^i honest one-hop neighbors. Note that falsely accused nodes do not directly pose a security threat. However, besides the inconvenience false accusations may cause, a large number of falsely accused nodes could stop the revocation scheme from working efficiently or in the worst case from working at all. To be able to revoke keys of

nodes in l-hop distance, with $1 < l \leq m$, a node i must receive at least δ (direct or reported) accusations from trusted nodes. Each node i trusts all its one-hop neighbors $n \in \Theta_i = (H_i \setminus F_i) \bigcup U_i$, which follows that the number of trusted one-hop neighbors is $|\Theta_i| = (1-\alpha)n_h^i + \beta n_m^i$. If $|\Theta_i| \geq \delta$, one-hop neighbors are able to cause the revocation of a key per direct accusations. On the other hand, for $|\Theta_i| < \delta$, node i must be able to accept reported accusations in addition to direct accusations. In that case $|\Theta_i| \geq \varepsilon$ must hold. Note that only reported accusations can cause revocations of keys of nodes that are more than two hops away. From our discussion, we conclude that false positive rate α must satisfy the following inequality

$$\frac{n_h^i + \beta n_m^i - \varepsilon}{n_h^i} \geq \alpha,$$

to allow the acceptance of reported accusations. To allow direct accusations the bound can be relaxed by replacing ε by δ in the equation. The derived bound serves only as rough guideline and to keep inconvenience to a minimum and allow efficient functionality of the revocation scheme, α should be selected as small as possible.

5.4.5 Colluding One-hop Neighbors

After showing the impact of false positive rate α on the revocation scheme's security and functionality, we now analysis the resilience of the scheme to colluding one-hop neighbors. In particular, we show how security parameter δ and ε, as well as false negative rate β should be chosen to prevent such attacks.

Def. *Successful Attack by Colluding One-hop Neighbors*: A group of $n_c \leq \sigma_i$ colluding one-hop neighbors can convince an honest node i to mark the key of another honest node $j \in N_{m,i}$ as revoked in $\mathcal{KRL}^i$, i.e. $X_j^i = 1$.

An employed monitoring scheme with false negative rate β leads to $n_u^i = \beta n_m^i$ undetected malicious nodes in i's one-hop neighborhood. Hence, up to n_u^i one-hop neighbors $u \in U_i$ may collude to launch an attack. Colluding one-hop neighbors u can launch two types of attacks: *A. Altering direct accusations*, i.e. nodes u alter their own accusations as part of their propagated neighborhood watch messages, and *B. Altering reported accusations*, i.e. nodes u alter reported accusations of

nodes that are 2 to m hop away in their propagated update messages. We describe both attacks in the following and show how they can be prevented by appropriately selecting security parameters and monitoring scheme.

A. Altering direct accusations:
Recall that δ accusations revoke a key (see Condition 3 in Eq. (5.3)) and that node i directly copies the neighborhood observations of all trusted one-hop neighbors $j \in \Theta_i$ (see Step 4 in Revocation Algorithm 4). We now consider the following attack by colluding nodes $u \in U_i$:

- each node $u \in U_i$ sets $a^u_{j,u} = 1$ for an honest node $j \in N_{m,i}$ and sends a neighborhood watch message

Upon receiving $nm_{u,i}$, i uses u's neighborhood watch vectors $\underline{c}^u_u$ to update $\underline{c}^i_u$ in $\mathcal{KRL}^i$. In that way, node i updates n^i_u vectors in its revocation list, each containing $a^i_{j,u} = 1$. Thus, there are at least n^i_u accusations against node j in $\mathcal{KRL}^i$. Hence, if the following inequality

$$n^i_u \geq \delta$$

holds, node i will revoke node j's key. This result is not surprising because δ is the threshold for our revocation scheme, and thus δ malicious undetected one-hop neighbors can revoke the key of any $j \in N_{m,i}$.

The described attack can be prevented by selecting δ and β such that $n^i_u < \delta$. We know that $n^i_m \leq \sigma_i$, thus attacks by altering direct accusations can only succeed if $\beta\sigma_i \geq \delta$. Hence, if we select β and δ such that the following inequality

$$\beta < \frac{\delta}{\sigma_i} \tag{5.9}$$

holds, the described attack is completely prevented. We can observe that by selecting $\delta \geq \sigma_i$, Eq. (5.9) always holds because $\beta < 1$ for any monitoring scheme. In most monitoring schemes β typically ranges from 0.01 to 0.1 and thus δ can be selected smaller than σ_i. Hence, attacks by colluding one-hop neighbors altering their direct accusations can be prevented by selecting δ and β such that Eq. (5.9) holds, which does not put many restrictions on the parameter selection.

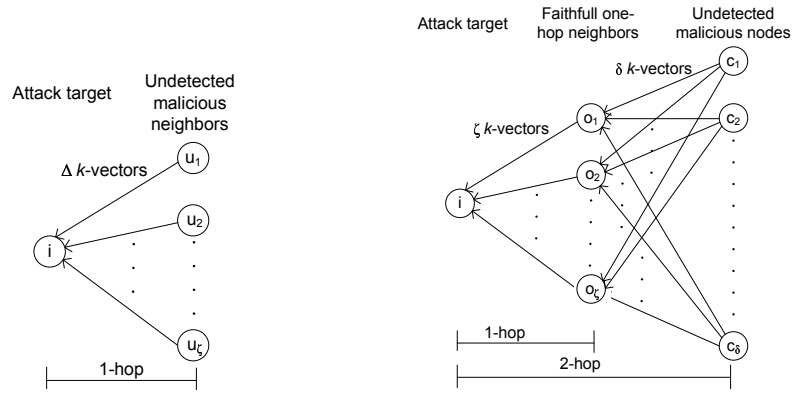


Figure 5.5: Attacks by Colluding Nodes: (a) One-hop Neighbors Altering Reported Accusations, (b) Two-hop Neighbors Altering Direct Accusations.

B. Altering reported accusations:
We now consider another attack by colluding one-hop neighbors $u \in U_i$, in which the adversaries alter reported accusations. Here, nodes u lie about accusation values of nodes that are more than one hop away. The attack exploits the majority rule (see Eq. (5.7)) that is applied by node i to derive column vectors $\underline{c}_k^i$, with $k \in N_{m,i} \setminus N_{1,i}$. From Steps 5 and 6 in Revocation Algorithm 4, we can observe that at least $\lfloor \frac{r_k}{2} + 1 \rfloor$ trusted nodes are necessary to gain majority and thus determine the accusation values in $\underline{c}_k^i$. In the attack described in the following, colluding nodes manipulate their submitted k-vectors. The colluding nodes $u \in U_i$ execute the following steps to launch an attack of type B:

- each $u \in U_i$ selects Δ nodes in $N_{m,i} \setminus N_{1,i}$, which is denoted as V, i.e. $V \subset N_{i,m} \setminus N_{1,i}$ and $\Delta = |V|$.
- node u sets $a_{j,v}^u = 1$ for all $v \in V$, and sends an update message $um_{u,i}$.

Upon receiving all n_u^i update messages $um_{u,i}$, node i updates the accusation value $a_{j,v}^i$ as $a_{j,v}^i = 1$ for all $v \in V$ if the received k-vectors from the colluders form the majority, i.e. if $n_u^i > \frac{r_k}{2}$ (see Eq. (5.7)). If the number of accusations reaches threshold δ, i revokes j's key with $X_j^i = 1$. Note that the minimum number

of colluding nodes to force the acceptance of the reported accusations is $r_k = \varepsilon$. Hence the minimum number of colluding nodes $u \in U_i$ is $n_u^i = \lfloor \frac{\varepsilon}{2} + 1 \rfloor$. In the remainder of our security analysis we use $\zeta = \lfloor \frac{\varepsilon}{2} + 1 \rfloor$. The attack is illustrated in Fig. 5.5-(a) and we summarize the conditions for a successful Attack B below:

1. all $v \in V$ are trusted by node i, i.e. $X_v^i = 0$ for all $v \in V$
2. $\Delta \geq \delta$
3. $n_u^i \geq \zeta$

We assume that colluders can easily ensure that Conditions 1 and 2 are satisfied. We now analyze how Condition 3 and thus a successful attack can be prevented by the proper choice of security parameters and monitoring scheme. Recall that $n_m^i \leq \sigma_i$. Hence if we choose β and ε such that

$$\beta < \frac{1}{\sigma_i} \lfloor \frac{\varepsilon}{2} + 1 \rfloor, \tag{5.10}$$

the described attack is prevented. We would like to emphasize that Eq. (5.10) reflects the best possible scenario from the attackers' point of view, in which exactly $\lceil \frac{\varepsilon}{2} - 1 \rceil$ honest one-hop neighbors report k-vectors. Less or more honest nodes would both require a larger number of colluding nodes n_u^i, because $r_k < \varepsilon$ in the first case, whereas n_u^i must maintain the majority in the latter case.

Note that $1 \leq \varepsilon \leq \sigma_i$. Hence, selecting large ε relaxes the condition on the accuracy of the monitoring scheme, but it reduces the efficiency and functionality of our revocation scheme. However, in any case $\varepsilon \geq 1$ and thus selecting $\beta < \frac{1}{\sigma_i}$ ensures that Attack B is prevented for any selection of ε. Since σ_i varies for different neighborhoods an average value $\overline{\sigma}$ should be estimated for the network before selecting a monitoring scheme with appropriate β.

Remark 1. Colluding one-hop neighbors can combine Attacks A and B, i.e. alter direct and reported accusations. In that case only $\Delta = \delta - n_u^i$ k-vectors must be manipulated, because the colluders send n_u^i altered direct accusations. A combination of the described attack modifies Conditions 1 and 2 but not Condition 3.

Hence, such a combined attack can still be prevented by selecting β and ε such that Eq. (5.10) holds.

Remark 2. If System Assumption 4 does not hold (i.e. a node i does not know which nodes are in its one-hop neighborhood $N_{1,i}$), undetected malicious one-hop neighbors $u \in U_i$ could collude with undetected malicious l-hop neighbors $c \in C$, where $1 < l \leq m$ and $X_c^i = 0$, to launch the following attack. Colluders c share their identities and credentials with nodes u, such that one-hop neighbors u can fool node i into believing that those nodes are also one-hop neighbors. For instance, a node u can set its MAC and IP address to c's addresses (i.e. spoof node c) and then use c's key material to fool node i to believe that the message came from a node c that is one hop away. In this way the colluders force i to increase the number of its observed one-hop neighbors in $\mathcal{KRL}^i$ to $\sigma_i + |C|$. This in turn increases the number of undetected malicious one-hop neighbors $n_u'^i$ (real one-hop neighbors plus l-hop neighbors pretending to be one-hop neighbors). The one-hop neighbors can now launch an attack altering direct or reported accusations of $n_u'^i$ nodes. Note that the described attack requires the colluders to share their secret key credentials, whereas the other described attacks by colluders do not have this requirement. We conclude that this attack is prevented if System Assumption 4 holds or $n_u'^i < \delta$ for direct accusations and $n_u'^i < \zeta$ for reported accusations, respectively.

5.4.6 Colluding l-hop Neighbors

We now analyze attacks by colluding l-hop neighbors of i with $2 \leq l \leq m$.

Def. *Successful Attack by Colluding l-hop Neighbors, with* $1 < l \leq m$: A group of $n_c < \varrho_i$ colluding l-hop neighbors can convince an honest node i to mark the key of another honest node $j \in N_{m,i}$ as revoked in $\mathcal{KRL}^i$, i.e. $X_j^i = 1$.

A. Altering direct accusations:

We first consider an attack by colluding two-hop neighbors, i.e. $l = 2$, in which the colluders alter their direct accusations. These altered accusations are received by the one-hop neighbors of the colluders, which in turn report the (altered) accusations to node i. We assume the following attacking scenario:

- a group of colluding 2-hop neighbors C, with $|C| = n_c$ and $C \subset (N_{2,i} \setminus N_{1,i})$
- a group of nodes O that are one-hop neighbors of node i as well as of all nodes $c \in C$, i.e. $O \subset N_{1,i}$ and $O \subset N_{1,c}$ for all $c \in C$

Furthermore, we assume that all nodes $o \in O$ faithfully execute the revocation algorithms. The attack consists of two phases:

Phase 1.

- each $c \in C$ sets $a^c_{j,c} = 1$ and sends a neighborhood watch message
- each receiver $o \in O$ updates its $\mathcal{KRL}^o$ with $a^o_{j,c} = 1$ if $X^o_c = 0$ and revoke j's key if more than δ accusations were received

Phase 2.

- each $o \in O$ sends an update message reporting the altered accusations, i.e. $a^o_{j,c} = 1$ for all $c \in C$
- node i updates its $\mathcal{KRL}^i$ if it received at least $r_c > \varepsilon$ c-vectors. In that case the majority vote forces i to set $a^i_{j,c} = 1$ for all $c \in C$. And if the number of collected accusations is larger than δ, i revokes j's key

The attack is illustrated in Fig. 5.5-(b) and works if the following three conditions hold:

1. $|C| \geq \delta$
2. $|O| \geq \zeta$
3. all $o \in O$ mark each $c \in C$ as honest, i.e. $X^o_c = 0$ for all $o \in O$ and $c \in C$

We assume the first two conditions to be true and analyze Condition 3, i.e. the probability that each colluder $c \in C$ remains undetected by the monitoring scheme of each of its one-hop neighbors $o \in O$. For an easier representation and without loss of generality, we denote the one-hop neighbors as $O = \{o_1, o_2, \ldots, o_\zeta\}$ and the

colluding adversaries as $C = \{c_1, c_2, \ldots, c_\delta\}$. The probability that one adversary c_r with $r \in \{1, \ldots, \delta\}$ remains undetected by one neighbor o_s with $s \in \{1, \ldots, \zeta\}$ is

$$\frac{\beta n_m^{o_s}}{\sigma_{o_s}}.$$

The probability that c_r remains undetected by all considered ζ one-hop neighbors $o \in O$ is

$$\frac{\beta^\zeta n_m^{o_1} n_m^{o_2} \cdots n_m^{o_\zeta}}{\sigma_{o_1} \sigma_{o_2} \cdots \sigma_{o_\zeta}}.$$

Now each of the δ colluding attackers c_r must fool all one-hop neighbors $o \in O$, which leads to probability

$$\left(\frac{\beta^\zeta n_m^{o_1} n_m^{o_2} \cdots n_m^{o_\zeta}}{\sigma_{o_1} \sigma_{o_2} \cdots \sigma_{o_\zeta}} \right)^\delta.$$

Lets assume that all nodes o_s have approximately the same number of one-hop neighbors $\overline{\sigma}$ and approximately the same number of malicious one-hop neighbors $\overline{n}_m$, then the probability of a successful attack by δ colluding two-hop neighbors $c \in C$ is

$$\left(\frac{\beta \overline{n}_m}{\overline{\sigma}} \right)^{\delta\zeta}. \tag{5.11}$$

We know that $\overline{n_m} \leq \overline{\sigma}$ and $\beta < 1$, i.e. the term in brackets is smaller than 1. Furthermore, with typical values of β ranging between 0.01 up to 0.1, the probability of a successful attack becomes negligible for small β and larger exponents.

The described attack assumes that node i receives a total of ε c-vectors, where ζ contain the altered accusations. However, if node i does not receive any other c-vectors, the colluders must manipulate ε c-vectors and thus "convince" ε as opposed to ζ one-hop neighbors o, which further reduces the likelihood of the described attack.

B. Altering reported accusations:

To increase their chance of a successful attack, the colluders in 2-hop distance could alter reported accusations of neighbors in 3 to m-hop distance. In that attack, assuming the best possible case from the attackers' perspective, ζ instead

of δ colluders are sufficient to launch the described attack. However, with

$$\left(\frac{\beta\overline{n}_m}{\overline{\sigma}}\right)^{\zeta^2} \tag{5.12}$$

the probability of a successful attack is only slightly larger and still negligible.

We now argue that the described attack for $l = 2$ is the best possible attack for colluding nodes in l-hop distance, with $1 < l \leq m$. Colluders must always fool at least ζ one-hop neighbors. Then in the best possible case (from the colluders' perspective), the altered accusations propagate through the network. Consequently, the probability of a successful attack by l-hop colluders can never exceed the probability in Eq. (5.12).

5.5 Performance Analysis

The performance of the key renewal scheme is identical to the initial key generation and distribution algorithms, as described in the ID-based framework in Section 4.3.2. In the basic key renewal scheme, nodes must leave the network to obtain new key material from an external KGC. Hence, no communication or computational costs are imposed onto the network. In contrast, if a distributed on-line KGC is employed, k nodes acting as D-KGCs must collaboratively generate and distribute new keys. As in all schemes employing threshold schemes, e.g. [125, 85, 77, 33], the collaborative nature of communications increases the communication load of the network and the computational load of the selected k nodes. Note that key renewal in [124] only requires broadcast messages of the D-KGCs. However this scheme only allows key renewal of unrevoked keys, whereas our scheme allows key renewal after expiry and/or revocation, to allow nodes which keys have been compromised to re-authenticate to the KGC (or k D-KGCs) to obtain a new key.

The performance of the revocation scheme depends in large parts on the frequency accusation messages are sent, which in turn depends on the number of malicious nodes in the network. There are two possible approaches: 1) propagation of accusation messages are triggered by events, i.e. every time malicious behavior is

observed, key compromise detected or revocation lists updated, or 2) accusations are propagated periodically with period ΔT_A. First approach ensures fast propagation of accusations but increases the communication overhead when many accusations are reported. To avoid collisions and network congestions, nodes should wait for a random period τ after an accusation event occurs, before sending out accusation messages. In networks with high rate of malicious nodes, the second approach for propagating accusations is desirable. Here, nodes accumulate all received accusation messages for a period ΔT_A before propagating a summary of these messages. To decrease the number of collisions in this approach, nodes each install a timer that starts at a random time t_r and propagate accusations every $t = t_r + i\Delta T_A$ where $i \in \mathbb{N}$.

Another parameter that affects the network performance is propagation range m. Depending on the network load created by the revocation scheme, m can be adjusted. The smaller m, the lesser a message must be re-sent to the next hop. In general, each accusation message am_i is sent to $|N_{m,i}| - 1 = (\varrho_i - 1)$ nodes j. Thus small m decreases the network load. However, small m might cause that a node i that wants to communicate to a node j, has no revocation information about j, because j is outside of i's m-hop neighborhood $N_{m,i}$. In that case i must request revocation information from a node l in $N_{m,i}$ that has information about j, i.e. $j \in N_{m,l}$.

We now analyze the computational and communication costs of the different accusation messages. The computation of a harakiri message hm_i (see Eq. (5.5) is virtually free for node i and does not require any cryptographic operations. The message requires one broadcast to all nodes in range. The verification of a received harakiri message requires one pairing computation if receiver j holds a copy of $K_{j,i}$ or two pairing computations otherwise. Hence, the system costs of one harakiri message are at least $(\varrho_i - 1)$ pairing computations and at most $2(\varrho_i - 1)$ pairing computations. We conclude that the proposed harakiri messages are an extremely efficient way to revoke keys, especially when compared to the alternative approach of using digital signatures. Signature schemes require at least two computationally demanding computation steps for each verification, e.g. modular exponentiations.

A sender i of a neighborhood watch or update message $nm_{i,j}$, $um_{i,j}$, respectively, must compute $(\sigma_i - 1)$ MACs and up to $(\sigma_i - 1)$ pairing computations. Each receiver j of $nm_{i,j}$ or $um_{i,j}$ must verify the MAC and thus perform one pairing computation. Note that for senders as well as receivers, pairing computations are only necessary for pre-shared keys $K_{i,j}$ that have not been computed and stored yet. In a fairly static network, it can be assumed that both types of accusation messages only require MAC computations after an initial phase in which all pre-shared keys are computed.

Hence, we conclude that the computation costs of all accusation messages am_i are fairly low. Even in the worst case, i.e. a very dynamic network with frequently changing neighborhoods, our scheme is at least as efficient as revocation schemes using signatures, assuming that one pairing computation is not more demanding than one verification.

New nodes that join the network or move to a new neighborhood, immediately start their own neighborhood watch (Algorithm 1). After an initial observation time T_{init}, that may be used to establish routing information and other necessary set up tasks as well, node i obtains its first monitoring results about its one-hop neighborhood $N_{1,i}$. Node i can now start to use received accusation messages $am_{j,i}$ to update its key revocation list $\mathcal{KRL}^i$. Unlike all other accusation-based revocation schemes [32, 33, 67, 77, 85, 124, 125], our scheme does not require the verification of signed accusations. In our scheme, the protection of accusation messages, security parameters δ and σ, and the majority vote for computing k-vectors ensure the authenticity and accuracy of accusations. We can conclude that the procedure for newly joining nodes in our scheme is extremely efficient and does not impose any extra costs.

5.6 Discussions and Conclusions

In this chapter, we proposed a novel key revocation and key renewal scheme for pairing-based IBC schemes in MANETs. The proposed neighborhood watch scheme helps to detect malicious, selfish, and any other misbehaving nodes in MANETs

and revoke the keys of all detected nodes. All observations are securely propagated to an m-hop neighborhood. We provide a detailed descriptions of all algorithms and, as a novelty, our revocation schemes provides efficient and secure mechanisms for nodes to revoke their own keys and newly joining nodes to obtain past accusations. Our solution is applicable to any pairing-based IBC scheme in MANETs, including MANETs with distributed on-line KGC. For example, the proposed key revocation and key renewal schemes can be seamlessly integrated into the ID-based authentication and key exchange framework proposed in Chapter 4 as well as in the recently proposed IBC schemes for MANETs [77, 33], which do not provide neither of these mechanisms. Furthermore, the proposed revocation scheme can be adapted to PKI schemes in MANETs with off-line or on-line CAs.

In our extensive security analysis we show that our proposed key renewal and revocation schemes are secure and thwart many common attacks. In particular, we demonstrated that the proposed key renewal scheme thwarts Sybil and other impersonation attack, and that the key revocation scheme prevents attacks by outsiders, malicious non-colluding nodes and roaming adversaries. In our analysis of colluding attacks, we showed how security parameters δ and ε and system parameter β can be selected to entirely prevent attacks by colluding one-hop neighbors that alter their direct accusations (see Eq. (5.9)) or alter their reported accusations (see Eq. (5.10)). Furthermore, we demonstrated that the likelihoods of attacks by colluding l-hop neighbors are negligible and we show how δ, ε, and β can be selected to further reduce the propagation of directly altered accusations (see Eq. (5.11)) and altered reported accusations (see Eq. (5.12)).

In addition to its scalability using the security parameters, the performance and security of the revocation scheme can be adjusted with parameter m. For instance, greater m decreases the chances of roaming adversaries to remain undetected, where smaller values increase the scheme's performance with respect to bandwidth and memory space. Our solution is very efficient due to the use of pre-shared keys in MACs to secure accusation messages as opposed to using signatures. In addition, the solution has lower communication costs because messages are propagated to an m-hop neighborhood instead of to the entire network. Unlike existing solutions for

MANETs, our solution provides a very efficient way for nodes to revoke their own keys. Furthermore, newly joining nodes can simply join the network and start the revocation scheme without first verifying a large number of past accusations and revocations.

Chapter 6

Key Escrow Problem in MANETs

The KGC in IBC schemes is a key escrow because it knows all private and pre-shared keys used in the network. The inherit key escrow property of IBC schemes might be desirable in some cases, such as governmental and military applications, but in many other applications the property may be considered a drawback. In this chapter, we analyze the special role of key escrow in the context of MANETs. We show that by implementing IBC schemes in MANETs, we can benefit from the advantages those schemes have to offer while the impact of key escrow is minimized by the special properties of MANETs. We analyze the probability of successful key escrow attacks by malicious KGCs in MANETs and show countermeasures to either prevent attacks or reduce the probability of success. In addition to analyzing the prevention of key escrow in MANETs, we also study applications in which key escrow is desirable, e.g. to monitor network nodes. Therefore, we show how a KGC can increase its power as key escrow in MANETs. Hence, in this chapter we explore the two faces of key escrow.

For our analysis, we introduce two adversary models for dishonest KGCs that cope with the special properties of MANETs in Section 6.2. The fist model considers conventional KGCs, whereas the second introduces the new concept of so-called *spy nodes* that are distributed in the network. In the following section we review existing methods for key escrow prevention and discuss their applicability to MANETs. In Section 6.3, we discuss the necessary conditions for successful passive

and active attacks in each adversary model. Finally, we draw some conclusions in the last section. Earlier versions of our analysis and the proposed spy model can be found in [54, 59].

6.1 Related Work

The problem of key escrow in IBC schemes is known since the introduction of the schemes and has been studied for several years. Proposed solutions to prevent key escrow in IBC schemes consider implementations in traditional networks, i.e. static networks with an infrastructure and wired communication channels. In this section, we highlight some of the proposed methods and discuss which of these methods are applicable to MANETs.

Using a DH-like key agreement to prevent passive attacks by dishonest KGCs in IBC schemes is widely known and discussed in [27]. The approach has been applied to several ID-based AKE protocols, e.g. [19]. In the following Sections, we show how the method can be used in our adversary Models I and II to prevent passive attacks.

The general approach for preventing key escrow is to distribute the KGC's master secret s, to several entities, say n, such that at least k of those entities have to collude to place a key escrow attack. Examples of such proposed solutions are: (1) using a (k, n)-threshold scheme to distribute the master key [17]; (2) using n KGCs to issue partial private keys that are added up by users to obtain full private keys [28,51,101]; (3) using one KGC and $(n-1)$ key privacy agencies to sequentially issue private keys [81]; and (4) using a KGC and a mediator who each know a part of the users' private keys [99]. None of these solutions have been particularly been proposed for implementation in MANETs. We show in our analysis in Section 6.3, that the same methods can be used as a countermeasure to reduce the probability of a successful attack. The approach of multiple KGCs is suitable for MANETs because it only adds overhead to administrative tasks that are performed by the KGCs and does not affect the performance of communications among network nodes at all. After an initialization phase all communications and other activities among

the network nodes are the same as in implementations with a single KGC. However, in many applications users must accept a provider's terms and conditions and thus may have no choice but to trust a KGC or group of KGCs. Furthermore, KGCs that are part of a distributed KGC need to cooperate to set up the system, agree on parameters, algorithms, security policies, etc.. This cooperation during set up may suggest cooperation among several KGCs to enable key escrow. Hence, in our analysis we do not distinguish between single dishonest KGCs or groups of colluding KGCs.

Another well-known method that decreases the probability of master key compromise is to assign an expiration date to master secret key s [17]. However, the method only addresses honest KGCs that have been compromised and does not prevent a dishonest KGC from being a key escrow.

Additionally, the method proposed in [41] suggests the encryption of messages using additional private/public key pairs which are not known to the KGC. This approach is not suitable here because the additional private/public key pairs are not ID-based and thus require a PKI. This counteracts the reasons why we wanted to use IBC schemes and sacrifices the special features of IBC schemes that are attractive for deployment in MANETs.

6.2 Adversary Models For Dishonest KGCs

As the name implies, network users usually trust the system's trusted third party. However, this trust does typically not extend to the capability of this trusted third party to decrypt all protected communications in the network. For this reason, key escrow is considered a drawback in many network applications. In this section, we introduce adversary models for dishonest KGCs in IBC schemes that abuse their power as a key escrow to launch passive or active attacks on the users privacy. In particular, we consider scenarios in which a KGC is attempting to eavesdrop on communications between two nodes i and j in a MANET. In our analysis we consider three different mechanisms (fully described in the next section) that can be used by i and j to protect their communications in the network. We discuss the

attacks that KGC can launch and introduce two adversary models for dishonest KGCs in Sections 6.2.2 and 6.2.3, respectively.

6.2.1 Communication Protocols

Throughout our analysis, we consider node-to-node communications between two nodes i and j in a MANET with focus on privacy and authenticity of their communications. We do not consider lower layer protocols, e.g. we assume that secure routing is in place for multi-hop communications. We consider the following three types of ID-based protocols for protecting node-to-node communications:

Protection 1: Static Key Encryption. Nodes i and j use their long-term keys for encryption and decryption, i.e. pre-shared keys K_{ij} (see Eq. (4.4)) or public and private key pairs (Q_i, d_i) and (Q_j, d_j) (see Eq. (4.1) and (4.2)). Consequently, the same key is used to secure all communications which follows that once the key is compromised, an adversary is able to decrypt all previous and future communications.

Protection 2: Symmetric Key Exchange. Before starting to exchange data, i and j execute a symmetric ID-based AKE protocol, such as Protocol 5. We assume that the established session key SK is used to protect all subsequent communications between i and j during that session. As discussed in Section 2.2.3, PFS cannot be provided in symmetric AKE protocols.

Protection 3: DH-based Key Exchange. In the third scenario, i and j execute an ID-based AKE protocol in which session key SK is derived using a DH-based key exchange, such as Protocols 7, 8 and 10. Due to the use of a DH key exchange, the protocols achieve PFS.

6.2.2 Attacks

In our analysis we distinguish passive and active attacks by malicious KGCs and we define them as follows:

Passive attacks: In a passive attack, a KGC eavesdrops on communications be-

tween i and j. In addition, a KGC might use its knowledge of private and public keys (d, Q) and/ or pre-shared keys K_{ij} to decrypt the eavesdropped communications.

Active attacks: In an active attack, a KGC does not only eavesdrop on communications and decrypt them if necessary, but also intercept, create, modify and re-direct messages. However, we would like to emphasize the difficulty of intercepting messages in MANETs. While eavesdropping is a fairly easy task in wireless networks, preventing broadcasted messages from propagating through the network is relatively hard. In order to intercept messages, the KGC needs the ability to jam signals in a controlled manner without arousing suspicion of the neighboring (affected) nodes. However, in the adversary models introduced in this chapter we assume that the KGC has all those capabilities. Consequently, a KGC can abuse its powers to masquerade as another network node during an active attack. We would like to emphasize the power of such an active attack, because the KGC has knowledge of all the key material of any arbitrary node in the network, including the node the KGC attempts to impersonate.

An example of an active attack in an IBC scheme is a *KGC-in-the-middle-attack*. During this attack, a KGC "sits" in the middle of two nodes i and j that communicate using Protection 1, 2, or 3 described in the previous section. In a successful attack, the KGC masquerades as i to j and vice versa. After the protocol execution, i and j each share a session key with the KGC but believe that they share a session key with each other. The KGC can now read all encrypted communications between A and B by intercepting and decrypting the messages from the sender and then re-encrypting the messages using the key shared with the receiver. Nodes i and j are perfectly fooled and cannot detect the KGC in the middle that listens to their communications.

6.2.3 Adversary Models

We now derive two adversary models for dishonest KGCs in MANETs, taking the special properties of MANETs into account. The first model can be derived

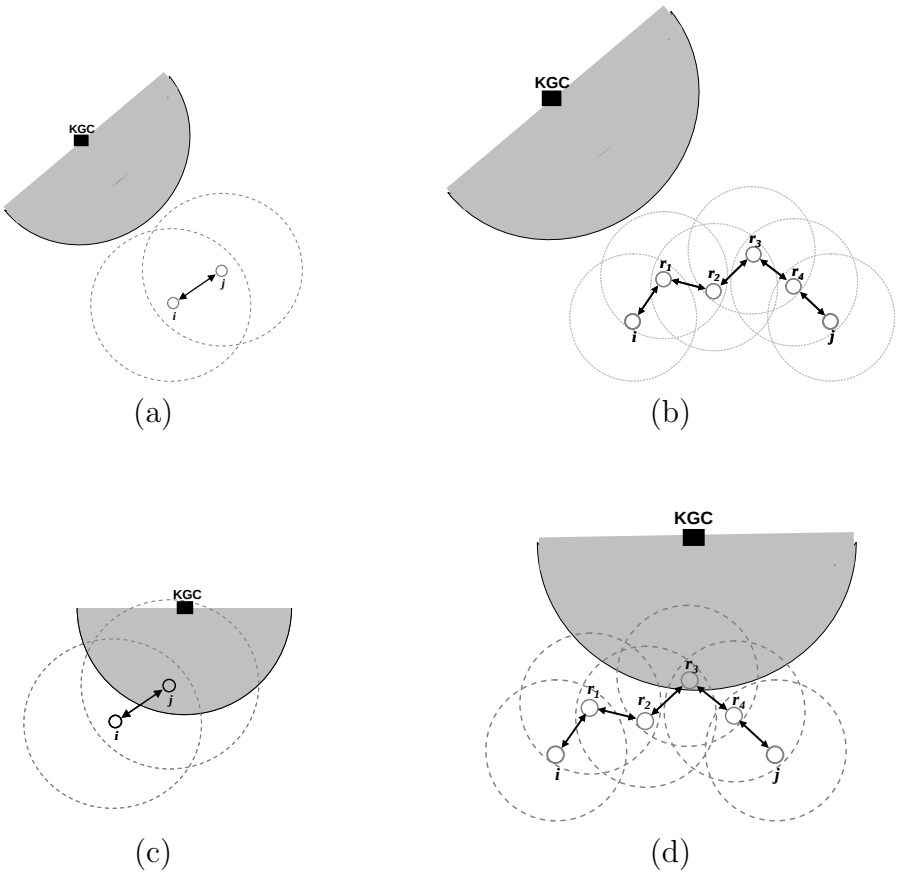


Figure 6.1: Model I: (a) one-hop communication and KGC outside communication range of i and j, (b) multi-hop communication and KGC outside the communication range of i, j, and all intermediate nodes r, (c) one-hop communication and KGC in communication range of i or j, here j, (d) multi-hop communication and KGC in communication range of i or j or at least one intermediate node r, here r_3.

intuitively for MANETs, whereas the second one is based on so-called *spy nodes*, a new concept that we introduce in this thesis. For each model, we consider one-hop and multi-hop communications among the network nodes (see Figure 2.1 (a) and (b)). The two models are illustrated for one-hop and multi-hop communications in Figures 6.1 (a)-(d) for Model I and Figures 6.2 (a) and (b) for Model II, respectively. In the figures, the communication range R of network nodes is depicted as dashed circle and the reception range R_{KGC} of the KGC as dark grey circle. We would like to point out that a KGC is usually very powerful and thus its transmission and reception range is much larger than the range of network nodes, i.e. $R_{KGC} >> R$.

Model I: Dishonest KGC model. In this adversary model we consider one KGC and several mobile nodes in a MANET. The KGC can be either outside the communication range of the communicating nodes i and j or inside their communication ranges. The first case, i.e. KGC is out of range, is illustrated in Figure 6.1 (a) for one-hop communications between i and j and in Figure 6.1 (b) for a multi-hop communications. Consequently, in one-hop communications the KGC is outside i's and j's communication range, whereas in the case of a multi-hop communication, the KGC is outside the communication range of i, j, and all intermediate nodes r on the routing path between i and j. In the second case, i.e. the KGC is within communication range, the KGC is either in direct communication range of i or j in the one-hop scenario (Figure 6.1 (c)) or inside communication range of i, j or any other node r on the routing path in the multi-hop scenario (Figure 6.1 (d)).

Model II: Spy nodes model. In our second adversary model, we introduce a new concept in which the KGC distributes so-called *spy nodes* in the network to increase its own communication range. We denote spy nodes as o in the remainder of this chapter. These spy nodes record all communications in their communication range and send the recorded data back to the KGC. Spy nodes have the following properties, they:

1. act and appear as regular network nodes
2. have the same power constraints as regular nodes

3. do not possess the master key s of the system
4. can send recorded messages to the KGC
5. can play messages received from the KGC back into the network

Spy nodes in our scheme have properties 1 and 2 for two reasons, namely in order to be cheap and to be indistinguishable from other regular network nodes. A spy model with more powerful spy nodes is briefly discussed in Section 6.4.1. Spy nodes cannot intercept messages because this requires jamming or similar capabilities, which is clearly beyond the power of a spy node. Spy nodes can use multi-hop routing to increase their limited communication ranges, e.g. to communicate with the KGC, where the routing paths may consist of spy and regular network nodes. By introducing spy nodes into the network, the KGC is able to eavesdrop on communications outside its own communication range as long as a spy node is in communication range, as illustrated in Figures 6.2 (a) and (b) for one-hop and multi-hop cases, respectively. In the figures, spy nodes o are depicted as grey circles and their communication range is shaded light grey. Note, that a spy node needs to be in communication range of i or j in the one-hop scenario and in range of i, j or one of the intermediate nodes r in the multi-hop scenario in order to record communications between i and j.

6.3 Analysis of Attacks and Countermeasures

In this section, we analyze the necessary conditions for successful passive or active attacks by dishonest KGCs in both adversary models. Furthermore, we discuss the likelihood of successful attacks. We show countermeasures for all attacks that can be prevented and, if a total prevention is infeasible, we explain how the probability of a successful attack can be reduced. We separately analyze passive and active attacks by dishonest KGCs on two communicating nodes i and j that use one of the three protection methods discussed in Section 6.2.1 to protect their communications. In our analysis, we make use of the following notations. We assume the communication range, i.e. transmission as well as reception range, of all network nodes and spy

nodes to be the same, i.e. $R = T_x = R_x$. Let R_{KGC} be the reception range of the KGC, with $R_{KGC} >> R$. Let $D_{i,j} = (i, r_1, r_2, \ldots, r_{l-2}, j)$ be a routing path between two communicating nodes i and j, where l is the length of the path. Furthermore, $|x - y|$ denotes the Euclidean distance between two points x and y. We use *session* for the execution of an AKE protocol and subsequent communications that are protected using the derived session key. The duration of a session is denoted as $T_{ses} = t_{s_1} - t_{s_0}$, where t_{s_0} is the starting time and t_{s_1} the time the session terminates.

6.3.1 Passive Attacks

Model I: If the KGC is out of communication range it cannot launch a passive attack, because the KGC is not able to eavesdrop on communications. However, if the KGC is in communication range of either i, j, or one of the intermediate nodes r, the KGC is able to eavesdrop on the communications between i and j. The conditions for a successful passive attack in Model I can be summarized as follows:

Condition 1. The KGC is in communication range of at least one node n_x on the routing path, i.e., $|KGC - n_x| \leq R_{KGC}$ with $n_x \in D_{i,j}$.

Condition 2. The communicating parties i and j use either no security protocol, or protection methods 1 or 2 for their communications.

Condition 3. If protection method 2 is used, the KGC needs to be in communication range during protocol execution.

The first condition is necessary for launching any attack. The second condition is necessary to enable the KGC to read/decrypt the exchanged messages. If i and j do not secure their communications at all, the KGC can simply eavesdrop on the communications. If protection method 1 is used, the KGC can directly decrypt the messages, whereas for protection method 2, the KGC needs to derive the session key first. For example, for deriving the session key in Protocol 5, the KGC needs to be in transmission range of the first and second flows (Condition 3).

Countermeasures:

AKE Protocol with PFS. Passive attacks by dishonest KGCs can be easily prevented by implementing a DH-like key agreement protocol or any other protocol

that provides PFS, such as Protocols 7, 8, and 10. By doing so, Condition 2 would not hold anymore. Only ephemeral public keys are exchanged during protocol execution, whereas computing session keys would require the KGC to know at least one of the ephemeral private keys. Hence, the KGC is not able to decrypt the communications anymore and is thus not longer a key escrow. Please note that using DH-type AKE protocols to prevent key escrow by passive eavesdroppers is a well known solution and not specific to MANETs [27].

Model II: In this model, the KGC cannot directly eavesdrop on communications between i and j. However, the KGC can use spy nodes to launch a passive attack. For a successful attack, Condition 2 and the following additional Conditions must hold:

Condition 1'. At least one spy node o is in communication range of a node n_x on the routing path to record the exchanged messages, i.e. $|o - n_x| \leq R$ with $n_x \in D_{i,j}$.

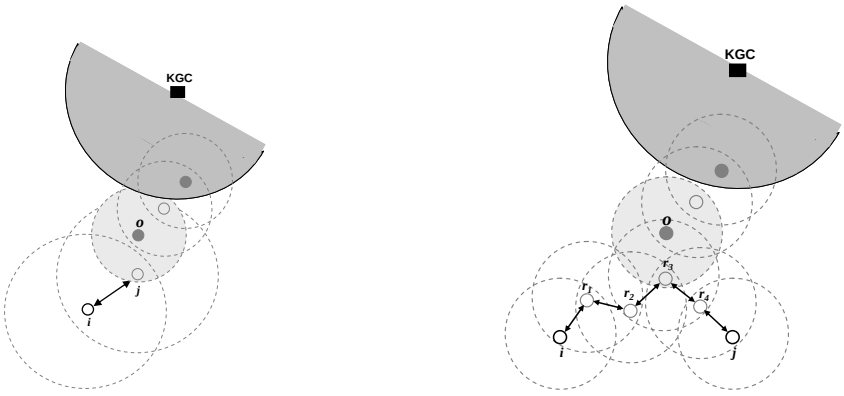
Condition 3'. If protection method 2 is used, at least one spy node o needs to be in communication range during protocol execution and record the first and second protocol flows, i.e. Condition 1' must hold for protocol flows 1 and 2.

In a successful passive attack, the recorded messages will eventually reach the KGC. The KGC is then able to decrypt the communications directly or derive the established session keys first and then decrypt.
Countermeasures: The same countermeasure as described for Model I can be applied, i.e. using an AKE protocol that provides PFS.

We do not analyze the likelihood of successful passive attacks because such attacks can be entirely prevented by the discussed countermeasure.

6.3.2 Active Attacks

Now we consider active attacks, where executing a DH-like key agreement or any other AKE protocol does not prevent the KGC from being a key escrow. The KGC could, for instance, launch a KGC-in-the-middle attack and derive a new DH key with each of the communicating parties without being detected. In the following



(a) (b)

Figure 6.2: Model II: (a) one-hop communication and at least one spy node o in communication range of i and j, (b) multi-hop communication and at least one spy node o in communication range of i or j or at least one intermediate node r_3.

paragraph, we derive conditions for each adversary model that are necessary for launching successful impersonation attacks which enable the KGC to read/decrypt communications, such as KGC-in-the-middle attacks. Next, we analyze the probability of successful active attacks in MANETs. Our analysis takes the dynamic topology of MANETs into account that is caused by nodes joining or leaving the network, as well as the mobility of nodes.

Model I: If the KGC is out of communication range, no attacks can be launched. The KGC can only launch an active attack on nodes i and j if the following condition holds:

Condition 4. The KGC is in communication range of at least two nodes n_x and n_y on the routing path between i and j during an entire session, i.e. $|KGC - n_x| \leq R_{KGC}$ and $|KGC - n_y| \leq R_{KGC}$ during T_{ses}, with $(n_x, n_y) \in D_{i,j}$.

If Condition 4 holds, the KGC can launch the following attack: the KGC intercepts the message from node n_x and then sends the modified message back to another node n_y that is on the routing path, where n_x and n_y do not need to be consecutive nodes on the routing path. The attack is illustrated in Figure 6.3 (a) for one-hop communications between i and j, here $n_x = i$ and $n_y = j$ with the

(a) (b)

Figure 6.3: Active Attacks in Model I with: (a) one-hop communication between nodes i and j, (b) multi-hop communication between nodes i and j.

KGC in the middle. In the multi-hop case, n_x and n_y are intermediate nodes on the routing path and the attack is illustrated in Figure 6.3 (b). The flash symbol in both figures symbolizes the jam signal or other mean of intercepting messages by the KGC. Nodes n_x and n_y are both not aware of the attack, because the KGC can simply masquerade as any other network node.

Probability of successful attack. After showing how a successful attack can be launched in Model I, we now discuss the likelihood of such attacks. Although we assumed that the KGC has the power to intercept messages, we would like to point out the difficulty of intercepting messages in MANETs. Even if we assume that the KGC has a way to jam specific signals without arousing suspicion and disturbing the actual protocol execution, we believe that a successful attack is still very unlikely for the following reasons. In order for an attack to be successful, Condition 4 needs to hold for each protocol flow u, i.e. for each flow of an AKE protocol and the subsequent communications. Recall that after a KGC-in-the-middle attack, the adversary (KGC) shares a key with i and another key with j. In order to prevent i and j from noticing the attack, KGC needs to continuously re-encrypt all messages with the respective keys.

We claim that the probability of a successful attack is low due to the dynamics of MANETs and the distance of nodes to the KGC as we argue in the following. Two packets of two different protocol flows u_x and u_y may not take the same route D_{ij} from a sender i to a receiver j, i.e. $D_{ij}(u_x) \neq D_{ij}(u_y)$. Hence, a dishonest KGC must be in communication range of at least two nodes n_x and n_y on route

$D_{ij}(u)$ of each round u. Lets say a protocol has U rounds, where routes $D_{ij}(u)$ with $u \in \{1, \ldots, U\}$ may be disjoint in different rounds, i.e. $D_{ij}(u_x) \bigcap D_{ij}(u_y) = \varnothing$ with $(u_x, u_y) \in U$ and $u_x \neq u_y$. Hence, the probability of a successful active attack, denoted by P_{suc}, depends on the probabilities $P_{N_I}(u)$ that the KGC is in range of at least two nodes n_x and n_y in each round u, i.e. $P_{N_I}(u) : \{|KGC - n_x| \leq R_{KGC} \wedge |KGC - n_y| \leq R_{KGC}; (n_x, n_y) \in D_{ij}(u), u \in (1, \cdots, U)\}$. Hence, the probability of a successful attack can be computed as

$$P_{suc} = \prod_{u=1}^{U} (P_{N_I}(u)). \tag{6.1}$$

Note that all discussed ID-based AKE protocols in this thesis have three flows, i.e. $U = 3$. We can observe from Eq. (6.1) that if $P_{N_I}(u) = 0$ in one of the (three) rounds, $P_{suc} = 0$, i.e. the attack fails. To derive the actual probability P_{suc} of a successful attack we need to calculate the probability P_{N_I}. Lets say N denotes all network nodes, where N_I contains all nodes inside the KGC's communication range and N_O all nodes outside the communication range, i.e. $N_I : \{|KGC - n_x| \leq R_{KGC}; n_x \in N\}$ and $N_O : \{|KGC - n_x| < R_{KGC}; n_x \in N\}$. The probability of having at least two nodes $(n_x, n_y) \in N_I$ on routing path D_{ij} cannot be easily derived, because the distributions of N_I and N_O are typically unknown and vary strongly for different applications. Typically the probabilities of having a node out of N_I or N_O on the path are not equal. In fact, we believe that $N_O >> N_I$, because most nodes are roaming in a large distance D to the KGC, with $D >> R_{KGC}$. Hence, the probability P_{N_I} of having a node out of N_I on the path is negligible which follows that the probability P_{suc} of a successful attack is negligible too in these scenarios. For an accurate analysis for particular applications, the distributions of N_I and N_O can be used.

However, we believe that our quantitative analysis is sufficient because the discussed scenarios are true for most MANET applications. For example, in a battlefield where the KGC remains in a safe place, whereas the nodes are placed far away in the enemy territory. Same is true in civil and other applications, where users do not necessarily roam close to the KGC any longer after network or node

initializations. The situation is comparable to PKI implementations, where nodes do not roam close to their CA any longer once they received their certificate. Hence, we believe that a successful attack is very unlikely due to the likely great distance between KGC and the network nodes.

Countermeasures: Even though we believe that the probability of an active attack is very small in Model I, we introduce the following methods to further decrease the probability.

Session control. As a countermeasure for all attacks in which the KGC can modify messages but cannot intercept them, we suggest that all network nodes r acting as routers discard received messages that belong to the same protocol flow but have different contents. For example, node r on the routing path should never receive two messages that appear to come from sender i, belong to the same protocol flow and session but have different contents.

Close proximity. Typically, the shorter the routing path, the less likely are two nodes out of N_I on the path. Hence, close proximity of nodes makes successful attacks very unlikely. For this reason, we suggest two nodes to establish a new shared key as soon as the nodes are in close proximity to each other.

Disjoint Paths. We suggest using different routing paths D_{ij} for packets whenever possible. One feature of MANETs is the redundancy of routing paths that offers several possible paths between a sender i and a receiver j. For most effective prevention, the used routing paths should be chosen to be completely disjoint for each flow, i.e. $D_{ij}(u_x) \bigcap D_{ij}(u_y) = \varnothing \; \forall \; (x, y) \in (1, \cdots, U)$, where $x \neq y$. In that case, probability $P_{N_I}(u)$ of each round u may be different which may reduce the overall probability P_{suc} of an successful attack. Note that if the KGC is outside the range in one of the rounds, the attack fails. If no disjoint or different routing paths are available, the network nodes should utilize their mobility to enable the use of different routing paths for different protocol flows.

Distributed KGCs using (k, n)-threshold or other schemes. A general countermeasure for key escrow attacks is the implementation of distributed KGCs as discussed in Section 6.1. However, as pointed out earlier, users have typically no choice which KGCs to select and KGCs must cooperate to establish the distributed

KGC, which may suggest cooperation for an attack.

Model II: Next we analyze active attacks in the spy model. An active attack similar to the one described for Model I is feasible in Model II, but here the KGC has a more realistic chance of a successful attack. In an attack, the KGC uses spy nodes to launch an impersonation attack on two network nodes i and j to ultimately eavesdrop on their communications. The attack is illustrated in Figure 6.4 and can be looked at as *spy-in the-middle-attack*. Unlike Model II, the KGC itself does not need to be in communication range and intercept messages. Instead, a spy node o on the routing path directly sends the exchanged messages to the KGC. Note that the spy node o does not need to intercept messages because it is a part of the routing path. The KGC modifies the messages it receives from o (after decrypting the messages if necessary) and sends them back to the spy node, which forwards the modified messages to the next node on the routing path. Please observe that spy nodes cannot launch the attack themselves, because they do not possess the necessary key material to impersonate network nodes. The communication between KGC and its spy nodes needs to be very fast in order for this attack to work. Otherwise the delay τ of a message would be too long and could cause the communicating nodes to drop the session and choose another routing path. The described attack is feasible if the following two conditions hold:

Condition 5. At least one spy node o is part of the routing path between i and j during an entire session, i.e., $o \in D_{ij}\ \{i, j\} = (r_1, r_2, \ldots, r_{l-2})$ during T_{ses}.

Condition 6. Spy node o and the KGC are able to communicate on-line, i.e. without long communication delays τ.

We can observe from Condition 5 that this kind of attack only works in multi-hop scenarios, because a spy o needs to be on the routing path between i and j in order to relay the messages to the KGC. In an one-hop scenario, a spy node would need to have jamming capabilities to intercept messages between i and j, which is beyond the capabilities of spy nodes. In order to avoid long delays τ between spy nodes and KGC, spy nodes could have a direct connection to the KGC using directed antennas, satellite connections or dedicated cables. In another realization, the routing paths between KGC and spy nodes are ensured to be short.

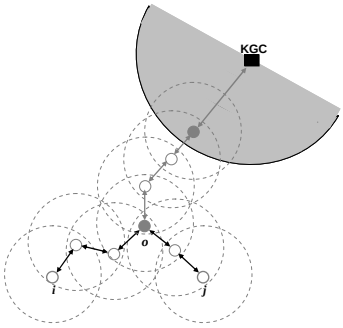


Figure 6.4: Active Attack in Model II with multi-hop communication between nodes i and j.

Probability of successful attack. For analyzing the attack we assume that Condition 6 holds. Hence, to calculate the probability of a successful attack P_{suc}, we need to determine the probability that Condition 5 holds, i.e. the probability P_{S_I} that at least one spy node o is part of the multi-hop path D_{ij} of length l. Note that here, the path excludes nodes i and j because they are obviously not spy nodes. The probability P_{S_I} depends on path length l and the distribution of spy nodes P_S and regular network nodes P_N in the network. The lengths l of routing paths depend on the used routing protocol, the number of nodes $\Omega = |N|$, mobility patterns, node locations, roaming area, communication range of nodes, and many other factors. Lets say we have a set or regular network nodes N, a set S of spy nodes, and l is the average length of routing path D_{ij} $\{i, j\}$ between i and j. The number of network nodes is denoted as $\Omega = |N|$ and the number of spy nodes as $\Psi = |S|$. For a successful attack on a AKE protocol with PFS, at least one spy node o needs to be a part of routing path D_{ij} of each protocol flow u. Hence, $P_{S_I} : \{o \in D_{ij}(u) \setminus \{i, j\} \; \forall \; u \in (1, \ldots, U)$ and $o \in S\}$ and the probability P_{suc} of a successful attack on a protocol with U rounds is given by

$$P_{suc} = \prod_{u=1}^{U} P_S(u),$$

where $P_S(u)$ might be different in each round u because $D_{ij}(u)$ and l might vary.

In the following paragraphs, we analyze the probability of an successful attack in some specific scenarios. We assume that spy and networks nodes are uniformly distributed in the network. Recall that the routing path $D_{ij} \setminus \{i, j\}$ consists of l nodes. The probability $P_S(l)$ of the event that we take l nodes out of the total set of all nodes, i.e. $\Omega + \Psi$, in which at least one node is a spy node is given by

$$P_S(l) = 1 - \left(\frac{\Omega}{\Omega + \Psi}\right)^l. \tag{6.2}$$

Eq. (6.2) describes the probability that one node the routing path is a spy node. We can observe that by increasing the number of spy nodes Ψ in the network, the KGC increases the probability of a successful attack, whereas decreasing the length l of the path decreases the chance. Please note that we discuss the attack for the case that exactly one spy node is on the path. However, if more than one spy node is on the path, they will each execute the attack, since spy nodes do not know of each other. Multiple spy nodes executing multiple attacks do not prevent the attack from succeeding, however, it would increase the communication delay τ.

We now compute the probability of a successful attack for some specific network scenarios. In our examples, we consider three-round protocols, i.e. $U = 3$, and assume that length $l(u)$ is the same in each round u. The probability $P_S(u)$ that at least one spy node o is on a routing path $D_{ij} \setminus \{i, j\}$ can be computed from Eq. (6.2). In Figures 6.5-(a) and (b), we illustrate P_S and P_{suc} for a network size of $\Omega + \Psi = 100$ nodes, where the number of spy nodes Ψ ranges from 0 to 100 and the average routing paths length is $l = 4$, 5, or 6. Note that the average path lengths in AODV routing protocols in networks of this size is $l = 4$ [118]. We can observe that for 5% spy nodes in the network, i.e. $\Omega = 95$ and $\Psi = 5$, the probabilities are $P_S = 0.1854$ and $P_{suc} = 6.38 \cdot 10^{-3}$, and for 10% spy nodes, i.e. $\Omega = 90$ and $\Psi = 10$, the probabilities are $P_S = 0.3439$ and $P_{suc} = 0.0406$. We believe that is reasonable to assume that 5%-15% of all nodes are spy nodes, because deploying more spy nodes is not cost effective for the KGC/adversary, whereas deploying less spy nodes seems too insignificant to increase key escrow power.

In a second example we consider a network size $\Omega + \Psi = 1000$, and probabilities

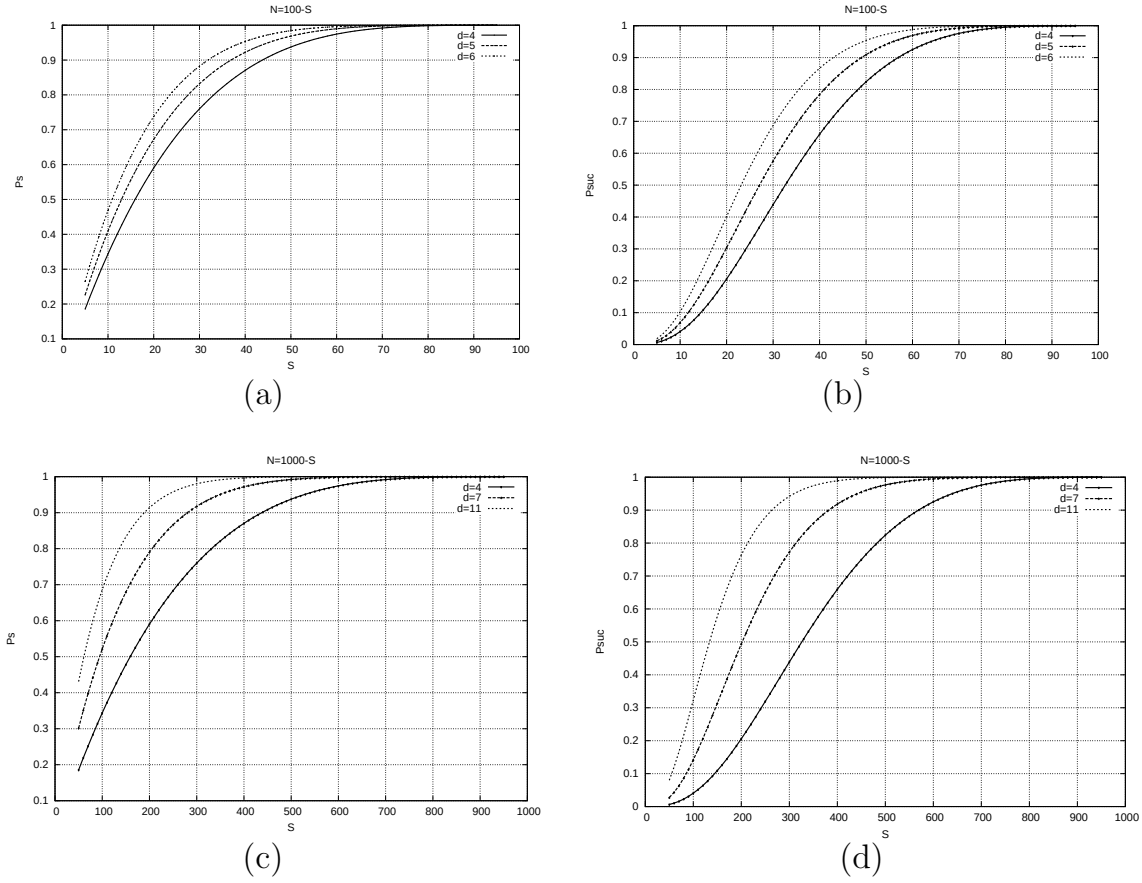


Figure 6.5: Probabilities $P_S(l)$ and P_{suc} for Varying Numbers of Spy Nodes S: (a) $P_S(l)$ for network size 100 and average path lengths $l = 4$, 5, or 6; (b) P_{suc} for network size 100 and $l = 4$, 5, or 6; (c) $P_S(l)$ for network size 1000 and $l = 4$, 7, 11; (d) P_{suc} for network size 1000 and $l = 4$, 7, 11.

P_S and P_{suc} are illustrated in Figures 6.5-(c) and (d) for varying number of spy nodes Ψ and average path lengths of $l = 4$, 7, and 11. Note that $l = 11$ in AODV routing protocols used in MANETs of this size [118]. We can observe that for 5% spy nodes, i.e. $\Omega = 950$ and $\Psi = 50$, the probabilities are $P_S = 0.43$ and $P_{suc} = 0.08$; and for 10% spy nodes, i.e. $\Omega = 900$ and $\Psi = 100$, the probabilities are $P_S = 0.686$ and $P_{suc} = 0.323$. This follows that if there are 5% spy nodes in the network, the probability of a successful attack is below 1% for a network size of 100 and significantly below 10% in a network of 1000 nodes. Even in the extreme case with 10% spy nodes, the probability is below 5% in the smaller network and around 32% in the larger one. The probabilities as illustrated in Figures 6.5-(a) to (d) serve as a rough estimate and probability P_{suc} highly depends on the average path length l which in turn depends on the efficiency of the implemented routing protocol and the mobility of the nodes.

Countermeasures: We showed that the probability of a successful attack is fairly small. However, the probability can be further reduced by one or more of the following countermeasures.

One-hop communications. Recall that active attacks in this model are only feasible in the case of multi-hop communications. As a consequence, one-hop communication can completely eliminate active attacks by dishonest KGCs. Hence, we suggest that two nodes establish a fresh shared key whenever they are in direct communication range.

Close proximity. Even if direct communication cannot be provided, close proximity between communicating nodes results into shorter routing paths, which in turn significantly reduces the probability of a successful attack. For this reason we suggest to take advantage of these events and derive a session key whenever the distance between two nodes is small.

Delay detection. Communicating nodes can check the delays τ of their protocol flows and if a flow takes more time than an estimated delay τ_{est}, i.e. $\tau > \tau_{est}$, the session is dropped. In that case, a new protocol run can be initiated using different routing paths. Especially if several spy nodes are on the path, the communication delay τ may be fairly large, and thus easy to detect. Please note that dropping a

session after a certain timeout period is a common practice in many protocol implementations, independent if the delay is caused by an attack, the communication channel, or other non-security related reasons.

Distributed KGCs using (k, n)-threshold or other schemes. The use of multiple KGCs to distribute the power was described as a countermeasure in the previous adversary model and is applicable to this model as well.

6.4 Monitoring Network Nodes

As mentioned earlier, in some applications key escrow might be a desirable feature. For instance, in some networks, the network provider might be interested to monitor network users for some legal issues. At the time users sign up for a service, they agree that the provider is able to monitor their communications in the network. However, communications or provided services are secured and can only be monitored by one party, the KGC, which might be operated directly by the network provider. In some other applications, such as government, military, and law enforcement applications, users might not be aware that the KGC can monitor their communications. Our analysis helps to understand what a KGC needs to do to maintain the key escrow property in MANETs.

As we pointed out in Model I, due to short communication ranges of wireless mobile devices and mobility of users, the probability P_{suc} of successfully monitoring two nodes i and j is very low. Note that the probability of a successful attack of a dishonest KGC is the same probability as for successfully monitoring nodes. In applications where key escrow is desirable, P_{suc} must be maximized. From our analysis we can observe that in a regular network without spy nodes, i.e. Model I, the probability of successfully monitoring nodes is negligibly small. Hence, the spy node model, i.e. Model II, should be used when monitoring nodes. We observe from our results for Model II that P_{suc} can be increased by increasing the number of deployed spy nodes Ψ. However, we believe that it is not cost effective for a KGC to place more than $5 - 10\%$ spy nodes in a network. In addition, such an implementation does not scale well. From our analysis for the spy node model we can also observe,

that with longer routing paths, the probability of successfully monitoring nodes increases significantly. As longer routing paths occur in large networks, monitoring is potentially easier in such scenarios. Furthermore, the topology of the network, the implemented routing protocol and many other factor can influence the routing path lengths. All these factors can be used to the advantage of the KGC. We would like to point out that in fairly static MANETs, the probability that a spy node is on the routing path, i.e. $P_S(u)$, can be estimated to be the same in each protocol flow u. In that case the probability for successful monitoring is $P_{suc} = P_S(u)$. In summary, we can conclude that the introduced spy model significantly improves the key escrow capability of a KGC in MANET applications.

6.4.1 More Powerful Spy Nodes

In the presented scheme we assume that the spy nodes have the same capabilities as regular networks. However, in more advanced implementations spy nodes might be more powerful devices, and, for instance, have larger communication ranges, be equipped with directed antennas, or share dedicated may be even wired channels with the KGC. Furthermore, the KGC or the provider of the network might choose a more sophisticated strategy to place spy nodes in the network, e.g. at natural or artificially created bottlenecks in the network. This would significantly increase the probability that messages are routed through spy nodes placed at these locations. In our analysis we assumed secure routing protocols, however without such protocols, spy nodes can advertise that they are on the shortest path to the destination even if they are not. KGC or network provider may be able to exploit other routing attacks on unsecured routing protocols to increase the probability of successful key escrow.

The discussed measures can help a KGC to monitor nodes or increase the likelihood of a successful key escrow attack for the price of more expensive equipment and deployment costs.

6.4.2 Other TTPs

Dishonest TTPs of other security schemes, such as CAs in PKIs, can also use spy nodes to increase their power to launch attacks in MANETs. However, the power of TTPs in other schemes is typically more limited than the power of KGCs in IBC schemes. For example, KGCs know the private keys and pairwise secret keys of all network nodes, whereas CAs are able to issue public key certificates but are not aware of secret or private keys of users. Hence, while the presented adversary models are applicable to other schemes, such as PKIs, the described attacks in this chapter are specific to IBC schemes. For instance, CAs can never launch passive attacks, independent of which protection mechanism is used to protect communications (Protection 1, 2 or 3 in Section 6.2.1). A CA can launch an active attack by generating key pairs (Q_i, d_i) and issuing false certificates $certi_i$ for the keys, e.g. for node i. The attack is detectable because multiple certificates exist for the same identity i but different public keys Q_i. Similar attacks cannot be detected in IBC scheme, because the KGC is in possession of the same key material than the impersonated node.

6.5 Discussions and Conclusions

In this chapter, we considered the special role of key escrow in MANETs which has never been studied in this context before. We introduced two adversary models of dishonest KGCs which take the limited communication range of MANET devices and multi-hop communications in such networks into account. We proposed a novel model in which so-called spy nodes are deployed by a KGC to increase its abilities to launch escrow attacks or legally monitor nodes. We were the first to explore enhancing key escrow capabilities to enable monitoring nodes in MANETs.

We showed that passive attacks can be prevented in all adversary models by using DH-like key agreement protocols. Active attacks by a dishonest KGC cannot be fully prevented neither in MANETs nor other networks. However, we demonstrated that the probability of a successful active attack is significantly lower in MANETs than in other wired networks with infrastructure. The results of our analysis re-

vealed that active attacks in Model I and Model II are only feasible under certain restrictive conditions. We evaluated the probability P_{suc} of successful active attacks in both models and showed that successful attacks are rather unlikely. We derived a formula to calculate P_{suc} in Model II and showed in Figures 6.5-(b) and (d) that the chance of a successful attack is less than 1% in networks consisting of 100 nodes in which 5% of all nodes are spies. Hence, the probability of successful attacks is much lower in MANETs than in traditional networks.

In addition to our analysis, we presented countermeasures to further reduce the likelihood of successful attacks. From our discussion, we conclude that the special properties of MANETs combined with the presented countermeasures prevent a KGC from being a key escrow in many MANET applications. On the other hand, we showed how a KGC could utilize spy nodes to monitor nodes and enhance its key escrow capabilities.

Chapter 7

Future Trends and Their Impact on Security Solutions

In this chapter, we will discuss some future trends of MANETs and analyze the impact of such trends on both existing security solutions and the design of new solutions for MANETs. In addition, we outline some security solutions for such envisioned applications. Currently, many proposed security solutions assume that no external TTP is available at any time (see AV-4 in Section 2.1.6 and Figure 2.2). However, we believe that this worst case scenario is more of academic nature and most real-world MANET applications (will) have access to some infrastructure such as the Internet, TTPs, backbone networks, etc. For example, we believe that the existence of a TTP to set up the network (AV-3) or initialize all nodes (AV-2) is a realistic assumption in most current and future MANET applications. We predict for the near future that network access and thus access to an infrastructure will become commonly available at many locations, e.g. via wireless access points (APs) and base stations (BS). Hence, in the future, network nodes will be able to at least sporadically access a TTP (AV-1) from within a MANET. In such applications, MANETs act as an extension to existing infrastructure networks. For example, nodes which are not in direct communication range of an AP to a network can use multi-hop routing to reach this AP. If at least one network node is in range, the other nodes in the MANET are connected to the infrastructure through this node.

The described approach is already deployed in wireless mesh networks (WMNs) in which nodes may access the wireless mesh backbone (WMBB)by forming a MANET to connect to one of the network's APs.

In this chapter, we use WMNs to illustrate remaining challenges of securing MANETs which act as extensions to existing infrastructure networks. We distinguish three categories of problems: (1) security problems of technologies that are currently used; (2) problem of efficiently and securely adopting solutions from MANETs; and (3) problem of efficiently adopting protocols that have been proposed for wired infrastructure networks.

We briefly address the first category in Section 7.1 as part of our overview of WMNs. We refer to [62] for our detailed discussion of security flaws of the widely deployed EAP framework that is used for client authentication in WMNs. We address problems of the second and third categories in more detail in Sections 7.2 and 7.3. More particularly, we modify the key renewal and key revocation schemes from Chapter 5 for use in WMNs, which significantly improves the performance of the schemes. In Section 7.3, we improve the efficiency of some AKE protocols that require sporadic infrastructure access.

7.1 Security Challenges in WMNs

We gave a brief overview of WMNs (infrastructure, client and hybrid WMNs) in Section 2.1.7 and now discuss security challenges of such networks. Currently deployed WMNs only support two wireless technologies, namely IEEE 802.11 and IEEE 802.16, where IEEE 802.11 is used for communications in the WMBB. Currently, the security offered by existing WMNs is based on the employed wireless standard, i.e. IEEE 802.11i or IEEE 802.16e. In general, communications within the WMBB are easy to secure due to stationary mesh routers (MRs) and the fact that the backbone is set up by one domain controller. Hence, pre-shared secret keys provide a suitable and efficient security solution. In addition, client access authentication is enforced by the employed wireless standard. Once clients have successfully authenticated, network access is granted and the clients may access the Internet

or other networks through the WMBB. EAP has been adopted as an access authentication and key establishment framework in IEEE 802.11i and IEEE 802.16e. For example, a mesh client who wishes to access a network starts an EAP session with a MR in range, where the MR passes the messages to an AAA server in the backbone. The established keys can then be used to protect the link between the client and MR. Even though EAP is adopted by wireless standards, some security vulnerabilities exist. Vulnerabilities are mainly due to the three-party communication model with weak physically protected MRs acting as authenticators, a protocol execution across different network layers and media links, as well as the backward compatibility of implemented ciphersuites. We summarize the security challenges and potential attacks on EAP and some particular EAP methods in [62].

We can observe that communications between clients and MRs as well as communications within the WMBB can be secured using existing security solutions offered by wireless standards. In addition, the backbone can be secured using standard solutions for wired networks. Hence, the only communication links that still require (new) security solutions are links among clients in client or hybrid WMNs. Securing these communications is very important because wireless links provide no physical protection and mesh clients are at high risk of compromise. The security goals for client to client communications in WMNs are the same as in MANETs, i.e. pre-authentication, authentication, and key exchange. However, like in MANETs, the mobility of clients, device constraints and the lack of pre-existing trust make providing such security goals difficult. For instance, it cannot be assumed that mobile clients pre-share any credentials, and thus existing solutions such as in IEEE 802.11i and IEEE 802.16e are not applicable. In addition, certificate/key revocation poses a major problem in all public-key solutions employed in WMNs. For example, OCSP could be used, such that mesh clients access the WMBB and request the status of particular keys from a server in the backbone. However, OCSP and other on-demand revocation schemes require mesh clients to access the backbone whenever they need to verify the status of a key/certificate. Hence, these methods are not suitable for hybrid and client WMNs. On the other hand, certificate revocation lists could be provided for download from a server in the backbone or pushed to

all clients. While this approach is feasible in all WMNs, the problem of identifying malicious clients remains. Especially if malicious mesh clients are out of range of access points to the network, revocation servers that reside in the backbone have no evidence that could justify the key revocation of such nodes. We conclude that suitable revocation schemes for WMNs should satisfy the following two conditions:

1. Revocation information can be downloaded from a server in the backbone.
2. Key revocations are based on accusations reported by mesh clients monitoring other clients in their neighborhood.

Considering the discussed security challenges and constraints of WMNs, it seems natural to adopt security protocols that have been proposed for MANETs to secure client to client communications in WMNs. For the same arguments as made for MANETs in Section 4.2, we believe deploying ID-based schemes to secure WMNs is desirable. Rather than simply adopting solutions that have been proposed for MANETs, such as our ID-based solutions in Chapters 4 and 5, solutions for WMNs should be modified to take advantage of the existing infrastructure to increase the performance and conserve energy of battery powered mesh clients. On the other hand, it is crucial to keep the introduced network load in the bandwidth constrained WMBB to a minimum. We discuss the impact of infrastructure in MANETs on our security solutions in Chapters 4 and 5 in the remainder of this chapter.

7.2 Efficient Revocation in WMNs

We propose using the ID-based security framework from Chapter 4 to secure client to client communications in WMNs and achieve the identified security goals. Recall that Algorithms 1-3 from the basic framework are executed by a central KGC. In WMNs this KGC could be placed in the backbone of the WMN. In addition, Algorithm 4, which establishes the pairwise pre-shared keys, is non-interactive and thus does not require any communication. Hence, Algorithms 1-4 of our ID-based framework do not need to be modified when deployed in WMNs. From our discussion in the previous section, we know that implementing key revocation and renewal

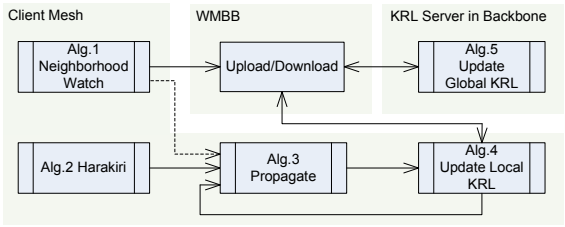


Figure 7.1: Overview of Key Revocation Scheme for WMNs

schemes are crucial, where first schemes should satisfy the two listed conditions. To provide such solutions, we propose key revocation and key renewal schemes for ID-based security schemes in WMNs that are based on our solutions in Chapter 5 and modified for an efficient deployment in WMNs by taking advantage of (sporadic) network access to the backbone. In our solution design we assume that all mesh clients in WMNs are able to at least sporadically access the WMBB and thus the backbone network. While this assumption is obvious for infrastructure and hybrid WMNs, we argue that clients in client WMNs can access the backbone prior to joining and/or upon leaving the network. Please note that in client WMNs in which clients never have access to the WMBB, the solution from Chapter 5 could be directly adopted without any modification. However, for all WMNs with sporadic access, solutions from Chapter 5 can be optimized as we present in the following subsections.

In our scheme, mesh clients upload their neighborhood observations to a central server in the backbone. The server generates a global key revocation list based on all received accusations using a threshold scheme. Mesh clients can download the global KRL whenever they need a fresh list and have backbone access. Only messages of higher priorities, such as harakiri messages are directly propagated to other mesh clients in m-hop range. An overview of the revocation scheme for WMNs is in Figure 7.1. We discuss the modified algorithms in the following sections, where we distinguish between algorithms for *key update* for keys that have been expired and *key renewal* for keys that have been revoked.

7.2.1 Key Revocation

As in the revocation scheme for MANETs in Section 5.3.1, keys are revoked whenever clients realize their own keys have been compromised (harakiri) or at least δ clients accused the same client. However, unlike the proposed solution for MANETs, observed behavior is not propagated through the network but rather reported to the KGC or another central server in the mesh backbone, referred to as revocation server *RS* in the remainder. The RS generates a global key revocation list $\mathcal{KRL}$ that can be downloaded by all nodes. Harakiri messages have higher priority and are thus still propagated through the network. As illustrated in Figure 7.1, the revocation scheme consists of five algorithms, namely *Alg. 1 Neighborhood Watch*, *Alg. 2 Harakiri*, *Alg. 3 Propagate*, *Alg. 4 Update local* $\mathcal{KRL}^i$, and *Alg. 5 Update global* $\mathcal{KRL}$ and a method to upload and download information to and from the RS. The upload/download method requires backbone access via an access point and messages are transmitted through the WMBB to the server in the backbone and vice versa.

The global key revocation list $\mathcal{KRL}$ can be represented as matrix

$$\mathcal{KRL}(t_S) = \begin{pmatrix} a_{1,1} & \cdots & a_{1,\Omega} & ID_1 & (t_{x_1}, v_1) & X_1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ a_{\Omega,1} & \cdots & a_{\Omega,\Omega} & ID_\Omega & (t_{x_\Omega}, v_\Omega) & X_\Omega \end{pmatrix}, \tag{7.1}$$

where we use the following notations: N denotes the set of mesh clients, $\Omega = |N|$ the total number of mesh clients, t_S the time the list was last updated, $a_{i,j}$ with $\{i, j\} \in N$ the accusation values which indicate whether node j accuses node i of malicious behavior ($a_{i,j} = 1$) or not ($a_{i,j} = 0$), ID_i with $i \in N$ the identity of client i, (t_{x_i}, v_i) the expiry date and version number of the current public key Q_i of client i, and X_i the revocation flag that indicates whether public key $Q_i(t_{x_i}, v_i)$ has been revoked ($X_i = 1$) or not ($X_i = 0$). Please refer to Chapter 5 for more detailed information on notations and parameter generations and computations.

Note that the revocation list can contain information of expired keys Q_i, with $t_{x_i} < t_S$ and stored keys may have different expiry dates t_{x_i}. $\mathcal{KRL}(t_S)$ is created and

initialized by the RS, and then updated and maintained using uploaded accusations from mesh clients. All accusation values and revocation flags are initialized with zero. Prior joining a client mesh network or whenever mesh clients have access to the backbone network and wish to update their revocation information, they download the global revocation list from the server. Each client i stores a local copy of the list, denoted as $\mathcal{KRL}^i$, which has the same format as the matrix in Eq. (7.1). In between downloads, clients i can update their local copies using their neighborhood watch observations and received harakiri messages. In the following we describe the algorithms of the modified revocation scheme for WMNs.

Upload/Download. In client mesh networks, client must download the global revocation list from the RS prior joining the network. If node i made observations that lead to modifications in $\mathcal{KRL}^i$, i should upload its entire list $\mathcal{KRL}^i$ or its neighborhood watch observations $\underline{c}_i^i$ to the RS upon leaving the network or at latest prior (re-) joining a WMN. To encourage uploads, the procedure can be combined with key updates, such that only clients which upload their accusations obtain fresh keys.

In hybrid and infrastructure networks, clients can download the list once they join the network. Here, clients should frequently upload their accusations to ensure accurate and timely key revocation lists. Uploads should be timed according to backbone accessability and network load. At the time a client i uploads its accusation values, i downloads the most recent $\mathcal{KRL}(t_S)$, where timestamp t_S prevents the distribution of old lists and unnecessary downloads. Again uploads should be encouraged by the network.

All communications between RS and i are secured by symmetric keys derived from a pre-shared key $K_{RS,i} = \hat{e}(sP, d_i) = \hat{e}(P, Q_i)^{s^2}$, where (s, sP) are the private and public key of the RS.

Algorithm 1: Neighborhood Watch. All mesh clients i monitor their one-hop neighborhoods $N_{1,i}$ for suspicious behavior and update their local key revocation list $\mathcal{KRL}^i$ accordingly (see Revocation Algorithm 1 in Section 5.3.1). If suspicious behavior is observed, the node sets its update flag ($update = true$), which indicates that the node will upload its observations to the RS at the next possible time.

Optionally, some special observations may have higher priority and trigger Algorithm 3 for a prompt propagation of the observations, as indicated by the dashed arrow from Algorithm 1 to Algorithm 3 in Figure 7.1.

Algorithm 2: Harakiri. The algorithm is identical to Revocation Algorithm 2 in Section 5.3.1, i.e. when a client i realizes that its private key d_i has been compromised, i generates a harakiri message $hm_{i,j}$ according to Eq. (5.5) and then starts Algorithm 3 to propagate the message to all its one-hop neighbors.

Algorithm 3: Propagate. Identical to Algorithm 3 in Section 5.3.1, i.e. harakiri and update messages are securely sent to all one-hop neighbors. This algorithm is triggered by Algorithm 2 and 4, but may also be triggered by Algorithm 1 for high-priority messages.

Algorithm 4: Update Local KRL. For received harakiri, neighborhood watch and update messages, the algorithm is identical to Algorithm 4 in Section 5.3.1, i.e. each client i uses received accusation messages to update their local revocation list $\mathcal{KRL}^i$, where revocation flags X_j^i are computed according to Eq. (5.3). Clients update their revocation lists every time they download a new copy of the global revocation list. If the downloaded list $\mathcal{KRL}(t_S)$ is newer than the stored local copy $\mathcal{KRL}^i(t_i)$, i.e. $t_S > t_i$, client i replaces all columns of its local list with the respective columns of the downloaded global list, except for the i-th column that contains i's own neighborhood observations.

Algorithm 5: Update Global KRL. The RS uses the uploaded accusations of mesh clients to update the global revocation list. Unlike in the MANET version of the revocation scheme, the server does not need to distinguish between one-hop and multi-hop neighbors. Instead, the server directly copies column vectors $\underline{c}_i^i$ from each trusted node i into column vector $\underline{c}_i$ in $\mathcal{KRL}$. The current public key $Q_i(t_{x_i}, v_i)$ is marked as revoked, i.e. $X_i = 1$, if at least δ accusations from trusted clients have been collected. In other words,

$$X_i = \begin{cases} 1 & \text{if} \quad \sum_k a_{i,k} \geq \delta \; \forall \; k \in N \text{ with } X_k = 0 \\ 0 & \text{else} \end{cases}$$

Every time an accusation value or revocation flag is updated, the server sets $t_S = t$ and provides $\mathcal{KRL}(t_S)$ for download.

7.2.2 Key Update

Mesh clients need to request a new key pair if their old keys are expired. Key updates require clients to communicate with the KGC in the backbone and can be performed at any time in infrastructure and hybrid WMNs, but need to be executed by clients prior joining client WMNs. In latter case, a registered user of a corporate WMN may download a new key pair when parking in the company's parking lot in which the gates are equipped with APs. In another scenario, users may download a new key pair from their desktop computers that are connected to the company's LAN onto their PDA before going to a business meeting. Many other examples of sporadic network access for key downloads are imaginable.

The frequency keys need to be updated, directly translates into the validity periods ΔT of public keys. ΔT should be chosen according to the accessibility of the network and the required security level. The KGC only issues fresh keys to clients whose most recent public keys $Q_i(t_x, v_i)$ have neither been revoked and nor been expired for too long, i.e. $t_x \leq t \leq t_x + \xi$ with $0 \leq \xi$. Parameter ξ is a security and performance parameter that indicates the grace period for key updates. During this extra time period after key expiry, a client does not need to re-authenticate because it can be assumed that the expired key has not been compromised since its expiration.

Upon receiving a key update request from client i at time t, the KGC verifies whether $Q_i(t_x, v_i)$ is marked as revoked in $\mathcal{KRL}(t_S)$ and $t_x \leq t \leq t_x + \xi$. If both checks are successful, the KGC generates a new private key $d_i(t_{x+1}, v_i = 1)$ with $t_{x+1} = t_x + \Delta T$ and sends it to client i. All communications are secured with keys derived from pre-shared key $K_{KGC,i}$.

7.2.3 Key Renewal

Key renewals are necessary whenever a public key $Q_i(t_x, v_i)$ has been revoked, i.e. $X_i = 1$, or keys have been expired for longer than ξ and client i wish. This algorithm is intended to help clients to recover from key compromises, e.g. as a result of accidently or maliciously revealed keys, key revocations that are based on false accusations, and missed key update deadlines. Hence, in contrast to key updates, clients need to fully re-authenticate to the KGC to obtain new keys. This is necessary to prevent adversaries which compromised a client to request new keys. Therefore, clients cannot use their revoked private keys d_i or pre-shared keys $K_{KGC,i}$ for re-authentication. Typically clients would re-authenticate using the same methods used to obtain the initial key, e.g. when registering for the network services.

7.2.4 Extensions

In addition to the extensions outlined in Section 5.3.4, the following modifications or extensions to our scheme for WMNs are possible:

- *Extended Neighborhood Watch.* Access points, base stations and mesh routers can be included in the neighborhood scheme algorithm to conduct accusation values based on their own observations of clients in communication range. Therefore, all access points to the WMBB do not only collect accusation values of clients in range but also upload their own accusation values to the RS. This modification improves the accuracy of the revocation scheme and ensures that the RS holds information about at least some mesh clients. Since access points are generally trusted more than mesh clients, their accusations could have more weight than accusations by mesh clients.

- *Local $\mathcal{KRL}$ Copies.* Access points can store a copy of the global revocation list $\mathcal{KRL}(t_S)$, such that mesh clients can download the list from these entities without the need to communicate with the RS in the backbone. Therefore, the RS frequently pushes the revocation lists to the access points. This extension

significantly reduces the communication load in the WMBB and allows much faster downloads for clients. Furthermore, access points can accumulate all uploaded accusations by clients over a certain time interval and forward them as one packet to the RS to further reduce the communication overhead.

- *Alternative Shared Keys.* Instead of using pre-shared key $K_{RS,i}$ to derive keys for securing communications between clients i and the RS, password or other pre-shared keys that have been established during the clients' registration to the WMN may be used. For example, pre-shared passwords and keys from implementations using RADIUS or DIAMETER servers, or freshly established keys from successful EAP authentications.

- *Faster Message Propagation.* It might be desirable to spread some accusation messages of high priority faster through the network. Like harakiri messages hm_i that are propagated to an m-hop distance, other messages could also be immediately propagated through the network in addition to uploading these accusations to the RS. However, this increases the network load, and if all accusation messages are propagated through the network the performance would be similar to the performance of the revocation scheme in Chapter 5.

7.2.5 Security and Performance Discussions

We refer to Section 4.5.1 for a security discussion of the ID-based framework, since the framework does not need to be modified for a deployment in WMNs. The security of accusation messages that are propagated through the client network, such as harakiri messages, is the same as in our revocation scheme for MANETs (see Section 5.4). All messages exchanged between each client i and the RS are secured with keys derived from the pre-shared keys $K_{RS,i}$ which provide message authentication, integrity and confidentiality. Hence, all messages are protected from attacks by outsiders.

Accusations are directly reported to the RS which maintains the global key revocation list. Keys are revoked if at least δ accusations from different trusted clients have been received. Hence, the revocation scheme is secure for up to $\delta -$

1 colluding insiders. The security of the key update algorithm depends on the likelihood of compromised pre-shared keys $K_{RS,i}$. Before private key d_i is expired, it is assumed that malicious behavior would be detected as part of the neighborhood watch algorithm. The likelihood of key compromise upon expiry depends on the time span between expiry t_x and key update request t, which is $t - t_x \leq \xi$. Hence, the likelihood can be adjusted with ξ. The security of the key renewal algorithms depends on the method used for client re-authentication.

The compromise of master secret s leads to a complete compromise of the network. However, we assume that KGC as well as the RS can be sufficiently protected since they are both located in the backbone. To further reduce the risk of compromise, the master secret s may be distributed over several entities [17, 19, 28, 51, 81, 99, 101].

The performance of the scheme depends on accessibility to the backbone and the frequency key revocation lists and accusations are downloaded and uploaded. In any case, the performance is significantly improved compared to the revocation scheme for MANETs in Chapter 5, because neighborhood watch and update messages are not propagated through the network. Hence, the overall computational and communication load is reduced. To avoid introducing a lot of additional communication load to the WMBB by downloads, we suggested storing global revocation lists on network access points and accumulating accusations before sending them to the RS. As in the original scheme, cryptographic pre-shared keys are used which enables the use of efficient symmetric cryptographic primitives.

7.3 Efficient Authenticated Key Exchange

Ideally, AKE protocols provide all required security properties while being efficient to comply with mesh clients' and network constraints. Despite the discussed advantages of IBC schemes in WMNs it might be desirable to implement conventional PKIs. For example, many public key schemes have already been standardized and are widely deployed. In addition, the key escrow property of IBC schemes might be considered as an obstacle for deployment. For these reasons, we discuss some

efficiency improvements to conventional public key-based AKE protocols in this section. The considered protocols use signed ephemeral DH keys to provide mutual authentication and authenticated key exchange. Our protocol is more efficient than a non-optimized signature-based AKE protocol and is thus of interest for computationally and power-constrained mesh clients and bandwidth constrained communication links. Many DH-based AKE protocols have been introduced and subsequently broken, see [18, 90] for a discussion. In this section, we analyze such an AKE protocol that has been broken and introduce a way to prevent the discovered attack without increasing the computational or communication complexity of the original protocol. We limit our discussions and comparisons on DH-like AKE protocols using digital signatures. Such protocols provide *(DP-8) non-repudiation* (see Section 2.2.3). However, more efficient DH-based AKE protocols that do not use digital signatures, such as the MQV and ECMQV protocols [2, 90], should be used if non-repudiation is not required.

In the remainder of this section we focus on efficient DH-based AKE protocols that provide the following security properties (as defined in Section 2.2.3):

- Mutual entity authentication (NP-1)
- Mutual implicit key authentication (NP-2), including key freshness
- Completeness (NP-3)
- Known-key security (DP-1)
- UKS resilience (DP-2)
- Key control (DP-3)
- KCI resilience (DP-5)
- PFS (DP-6)
- Non-repudiation (DP-8)
- Replay resilience (DP-9)

A common practice for achieving properties NP-2, NP-3, DP-1, DP-3, and DP-6 is using a DH key exchange, in which ephemeral DH-keys are exchanged to derive the session keys. However, ephemeral keys do not have public key certificates that bind owners and keys together and to provide key authentication, ephemeral keys can be digitally signed using long-term private keys. In addition to NP-1, digital signatures also provide properties DP-5 and DP-8. Properties DP-5 and DP-9 can be achieved in a generic way in three-round AKE protocols by using key confirmation and challenge and response techniques, respectively [18]. A naïve way to obtain an AKE protocol that provides all listed properties is using a signature-based AKE protocol (such as Protocol 3) with public key derivation. For instance, the Digital Signature Algorithm (DSA) [95] can be used to sign ephemeral DH keys. We refer to such protocols as *combined DH-DSA protocol*.

As a step towards more efficient DH-DSA protocols, Arazi proposed an *integrated DH-DSA protocol* [3] in which the DH key exchange is integrated into the DSA. However, Nyberg and Rueppel [98] demonstrated a known-key attack on Arazi's protocol. Recently, Harn et *al.* [50] proposed variants of Arazi's integrated protocol that prevent the known-key attack and provide resilience to replay and UKS attacks. Phan [119] pointed out that the protocols in [50] do not provide PFS and key freshness and adds these two properties in his protocol variant. However, the existing variants [50, 119] protocol add more security properties for sacrificing the efficiency of Arazi's original protocol. In this section, we introduce a new variant of Arazi's protocol that provides the same security properties as the most recent variant [119], while preserving the computational and communication efficiency of Arazi's original scheme.

7.3.1 Review and Analysis of Arazi's Integrated Protocol and its Variants

We use the following notation adopted from the DSA standard [95]: p is a large prime, q is a prime divisor of $p-1$, g is an element of multiplicative order q in $\mathbb{Z}_p$ and $h(\cdot)$ is a secure hash function. Mesh client i has a long-term private key

$d_i \in \mathbb{Z}_q$ and a long-term public key $Q_i = g^{d_i} \bmod p$. We assume that long-term public keys have been authentically exchanged prior to the protocol execution, e.g. using public key certificates. Hence, i and j are both able to compute a long-term secret key $L = g^{d_i d_j} \bmod p$.

The protocol flow of Arazi's protocol can be described as follows. Client i randomly chooses an ephemeral private key $k_i \in \mathbb{Z}_q$ and derives an ephemeral public key $m_i = g^{k_i} \bmod p$. The key m_i is then used to derive one part of the signature, i.e. $r_i = m_i \bmod q$. This step saves one modular exponentiation and constitutes the integration part of the protocol. The signature equation is solved in s_i according to DSA, i.e. $h_i = k_i s_i - d_i r_i \bmod q$ where $h_i = h(m_i)$. i sends (s_i, m_i) to j, which derives the second signature part r_i from m_i. In that way the communication complexity is reduced. j performs the symmetric steps when signing $h_j = h(m_j)$. After successfully verifying the received signatures according to DSA, both users compute the session key SK according to the DH key exchange, i.e. $SK = g^{k_i k_j} \bmod p$.

Given the protocol flow of Arazi's protocol, Nyberg and Rueppel [98] discovered that one can set up the following *attacking equation*:

$$SK^{s_i s_j} = g^{h_i h_j} L^{r_i r_j} Q_i^{r_i h_j} Q_j^{r_j h_i} \bmod p. \tag{7.2}$$

The only parameters not publicly known in (7.2) are SK and L and the equation can be solved in either SK or L by knowing the other parameter. We observe that known key resilience and PFS are compromised in Arazi's scheme because SK and L appear only as bases in the attacking equation. If instead the parameters would appear as bases as well as exponents in the attacking equation, solving the equations in either SK or L is no easier than solving the discrete logarithm problem [95]. Harn et *al.* prevent the known key attack in their integrated protocols by replacing h_i and h_j with $H_i = h(m_i||SK_{ij}||SK_{ji})$ and $H_j = h(m_j||SK_{ji}||SK_{ij})$, respectively, in their signature equations, where $K_{ij} = g^{k_i d_j} \bmod p$ and $K_{ji} = g^{k_j d_i} \bmod p$. In that way, the attacking equations for their protocols contain the session keys K_{ij} and K_{ji} as bases as well as inputs of the secure hash function $h(\cdot)$ which appear in the exponents of g. However, Harn et *al.*'s protocols do not provide PFS and key

freshness because, unlike in Arazi's original protocol, session keys SK_{ij} and SK_{ji} are not DH type keys. Phan adds PFS and key freshness to the protocols in [50] by computing both session keys SK_{ij} and SK_{ji} as DH type keys, with $SK_{ij} = Q_i^{k_i k_j} \bmod p$ and $SK_{ji} = Q_j^{k_i k_j} \bmod p$ where $n_i = Q_i^{k_i} \bmod p$ and $n_j = Q_j^{k_j} \bmod p$ are exchanged during protocol execution.

7.3.2 Efficient and Secure Integrated AKE Protocol

Our integrated DH-DSA protocol is based on Arazi's protocol and adopts the method of modified signing equations and message flows from the three-round protocol in [50]. Most parameters in our protocol, including the session key SK, are chosen and computed according to Arazi's protocol. In the first protocol round, client i computes ephemeral public key m_i and sends it to j. j in turn computes session key SK and ephemeral public key m_j, derives the signature part r_j and solves the modified signature equation

$$H'_j = s'_j k_j - d_j r_j \bmod q \tag{7.3}$$

in s'_j, where $H'_j = h(m_j||L||SK)$, i.e. the hash value is computed over j's ephemeral public key m_j, long-term secret key L, and session key SK. Then j returns (m_j, s'_j) in the second round. If i can successfully verify signature (r_j, s'_j), i computes SK and solves the modified signing equation

$$H'_i = s'_i k_i - d_i r_i \bmod q \tag{7.4}$$

in s'_i, where $H'_i = h(m_i||L||SK)$. i sends s'_i to j in the last protocol round and j verifies the received signature.

7.3.3 Security and Performance Analysis

The proposed integrated DH-DSA protocol prevents the known key attack by using modified signature equations (7.3) and (7.4). The proof is similar to the one in [50].

An adversary who observes a protocol execution can form an attacking equation

$$SK^{s'_i s'_j} = g^{H'_i H'_j} L^{r_i r_j} Q_i^{r_i H'_j} Q_j^{r_j H'_i} \bmod p. \tag{7.5}$$

The only unknowns in Eq. (7.5) are SK and L, however the equation can neither be solved in K nor L because both parameters occur as bases and as input of the secure hash function $h(\cdot)$ in the exponents of g. Hence, the proposed integrated protocol provides known key resilience and PFS. Note that Phan's attacks, as presented on Harn et. *al*'s protocols, are not feasible here because session key SK is a DH key, i.e. our protocol provides PFS and key freshness. Since our protocol uses a challenge-response structure and key confirmation it provides resiliency to replay and UKS attacks. Using digital signatures provides mutual entity authentication, KCI resilience and non-repudiation.

As in Arazi's protocol, each participant in our protocol needs to perform four modular exponentiations, namely one for computing the ephemeral public key, two for verifying the signature, and one for computing the session key. The only slight difference to Arazi's scheme are the longer input strings of hash values H'_i and H'_j, which have the same length as the hashes in the other variants [50, 119]. However, hash computations are extremely efficient compared to modular exponentiations and can be neglected for the overall computational performance. Furthermore, during their first communication two parties i and j need to compute long-term secret L which requires one modular exponentiation. We can observe that although our protocol has three protocol rounds, the same information as in Arazi's scheme and thus the same number of bits ($|p| + |q|$) are exchanged.

7.3.4 Alternative Crypto Schemes

The proposed integrated DH-DSA protocol was presented for a finite field implementation. However, the protocol can be easily modified to work in an elliptic curve group, i.e. integrating an ECDH key agreement into ECDSA. Note that an EC implementation of Arazi's original integrated DH-DSA protocol also suffers from Nyberg and Rueppel's attack because a similar attack equation as in Eq. (7.2) can

be derived, whereas our ECDH-ECDSA integrated protocol variant prevents the known-key attack without decreasing the protocol's efficiency.

In addition, the proposed integrated protocol can be implemented with second-order characteristic sequences using the LUC scheme [94,114] as well as third-order characteristic sequences using the GH scheme [46,47] or XTR scheme [83], respectively. The mentioned crypto schemes can be implemented using second-order or third-order linear feedback shift registers (LFSRs), respectively. Please refer to [45] for more information on LFSR sequences. In our initial work in [55] and [56], we showed that an attack equation similar to Eq. (7.2) cannot be derived for second- and third-order LFSR-based integrated DH-DSA protocols even if Arazi's original protocol is used. However, later is was discovered that in Arazi-like integrated protocols based on LUC or XTR schemes, an adversary can use the public key of one of the protocol participants to derive an attack equation with a probability of $\frac{1}{27}$ that has SK and L as only unknowns. However, this new attack is not feasible on integrated protocols from the GH scheme. Hence, integrated DH-DSA protocols employing LUC or XTR must use modified signature equations (see Eq. (7.3) and (7.4)) in order to prevent the discussed known-key attack. On the other hand, integrated DH-DSA protocols employing the GH scheme can be used without such modifications. In addition to the performance gain imposed by Arazi's integration step, LFSR-based integrated protocols help to further increase the performance due to their efficient implementation in hardware and use of smaller fields for computations than crypto schemes based on the exponential function, such as DSA. A detailed theoretical and experimental performance analysis is presented in [55].

7.3.5 Comparison

We summarize the performance and security properties of three-flow variants of the combined DH-DSA protocol, Arazi's integrated protocol, two three-flow variants of Arazi's protocol [50,119], and our proposed integrated protocol variant in Table 7.1. Here, we consider variants of Arazi's and combined DH-DSA protocols that are extended to three-rounds in a generic challenge-response manner to provide resilience to replay and UKS attacks. Note that this extension does not increase the com-

putational and communication costs. We only consider computationally expensive operations, i.e. we count the number of modular exponentiations (m.e.) and list the exchanged parameters and their bit lengths as a function of p and q.

We can observe from Table 7.1 that our protocol is the only protocol that achieves all considered security properties (1)–(10) while providing the same computational and communication efficiency as Arazi's original protocol. Our experimental results in [55] demonstrated that Arazi's integration steps improves the computational performance of a combined DH-DSA protocol by 20/which corresponds to our theoretical results and conforms that considering only computationally demanding operations is valid. Furthermore, our integrated protocol is the only one that prevents an adversary from obtaining the long-term secret key L from known session keys. Harn et *al.*'s protocol requires one additional modular exponentiation per user, resulting into the same computational performance as the combined DH-DSA protocol, and lacks key freshness and PFS. Phan's protocol achieves all security properties but requires two additional exponentiations per user and the exchange of $|p|$ additional bits per user, thus showing the worst communication and computational performance among all compared protocols.

7.4 Discussions and Conclusions

In this section we discussed the impact of sporadic or permanent infrastructure access on existing or newly designed MANET security solutions. We used WMNs as example to illustrate advantages and problems caused by such infrastructure access. Existing security protocols and standards can be used to secure such networks, but vulnerabilities may exist as discussed for the EAP authentication frameworks in [62]. We outlined a key revocation scheme, key renewal and key update schemes, and efficient AKE protocols all targeted to constrained environments in which nodes have at least sporadically the chance to access a network, servers or other kinds of infrastructures. The discussed solutions are only briefly outlined and subject to future research.

Our proposed ID-based framework from Chapter 4 in combination with the

	Combined DH-DSA	Arazi [3]	Harn et *al.* [50]	Phan [119]	Proposed protocol
Computation Costs/ User	5 m.e. *	4 m.e.	5 m.e.	6 m.e.	4 m.e.
Communication Costs/ User	(m, s, r) $\|p\| + 2\|q\|$ †	(m, s) $\|p\| + \|q\|$	(m, s) $\|p\| + \|q\|$	(m, n, s) $2\|p\| + \|q\|$	(m, s) $\|p\| + \|q\|$
Security Properties					
1. NP-1	X ‡	X	X	X	X
2. NP-2	X	X	– §	X	X
3. NP-3	X	X	X	X	X
4. DP-1	X	–	X	X	X
5. DP-2	X	X	X	X	X
6. DP-3	X	X	–	X	X
7. DP-5	X	X	X	X	X
8. DP-6	X	–	–	X	X
9. DP-8	X	X	X	X	X
10. DP-9	X	X	X	X	X

Table 7.1: Performance and Security Properties of 3-flow DH-DSA Protocols

* m.e. denotes a modular exponentiation in $\mathbb{F}_p$

† $|x|$ denotes the bit length of x, i.e. $x = \log x$

‡ X denotes that the security property is provided by the protocol

§ – denotes that the security property is not provided by the protocol

proposed key revocation and key renewal schemes can be used to secure communications in all types of WMNs. Communications between clients and servers in the backbone as well as among clients can be secured using the pairing-based pre-shared keys, where the revocation scheme allows parties to verify whether keys have been revoked. We showed how efficient key revocation can be provided in WMNs by taking advantage of sporadic backbone access to a central RS in combination with a local monitoring scheme. conventional revocation schemes are not applicable to WMNs due to the lack of both revocation information and permanent network access in hybrid and client WMNs. The proposed scheme is based on our revocation scheme for MANETs in Chapter 5 with significant performance improvements due to the use of global revocation lists.

Next, we introduced a new variant of integrated DH-DSA protocol that: (1) resists the known key attack that was proposed on Arazi's protocol; (2) provides all security properties of other Arazi-based protocols; and (3) preserves the excellent performance of Arazi's original protocol. Hence, our protocol preserves computational and communication performance without sacrificing security which makes our protocol as secure as the latest variant of Arazi's protocol [119] and as efficient as Arazi's original protocol. Using LFSR-based variants of the integrated protocol may lead to further efficiency improvements. Hence, the proposed protocols are suitable for constrained devices, such as mobile mesh clients, and bandwidth constrained communication links, such as the wireless links in WMNs.

Chapter 8

Concluding Remarks

In this thesis, we presented a complete ID-based security solution for MANETs including system set up, pre-authentication, authentication, key exchange, key revocation, key renewal and key escrow prevention. All proposed schemes are designed to meet the special security goals and constraints of MANETs, which were demonstrated in a security and performance analysis for each scheme. The basic versions of the proposed schemes can be employed in MANETs in which an external TTP initializes nodes before they join the network and the TTP might be still accessible at later times (see TTP availability scenarios AV-2 and AV-3 in Figure 2.2). In addition, we show how our solutions can be adopted to MANETs with no external TTPs (AV-4 in Figure 2.2) as well as to MANETs with sporadic infrastructure access (AV-1 in Figure 2.2). In the following sections we summarize the contributions of this thesis and outline directions for future work.

8.1 Summary of Contributions

- *Pre-Authentication Models:* We identified pre-authentication among nodes—that is, the initial establishment of pairwise shared credentials—as a necessary prerequisite for providing authentication, key exchange, and numerous other security goals in MANETs. We categorized several pre-authentication models that cope with the lack of infrastructure in MANETs, discussed ad-

vantages and shortcomings that helped to identify target applications, and presented solutions for each model. The models can be used to enable pre-authentication in the large number of existing security solutions for MANETs that previously neglected this security goal. As a result of our discussion, we identified self-authenticating public keys—and resulting non-interactive pairwise pre-authentication—of IBC schemes as attractive features to solve pre-authentication in MANET applications that previously had no sufficient security solutions.

- *Authentication and Key Exchange Framework:* We proposed an ID-based authentication and key exchange framework for MANETs that enables efficient and secure pre-authentication, authentication and authenticated key exchange among network nodes. The security of the framework is based on the underlying pairing-based IBC scheme and thus on the difficulty of solving the BDH problem. We described an algorithm for efficient system set up in which costs are solely carried by an external KGC. In addition, pre-authentication is efficient and secure because pairwise pre-shared keys are derived in a non-interactive fashion. Our framework is flexible and can be implemented with any pairing-based IBC scheme, where the derived pairwise pre-shared keys enable the use of any symmetric or ID-based AKE protocol to establish fresh session keys. The basic scheme is suitable for MANET applications with KGC availabilities AV-1, AV-2, or AV-3 (see Figure 2.2) and we outlined how the scheme can be modified to be completely independent of any external TTP (AV-4) by implementing a distributed on-line KGC using a (k, n)-threshold scheme.

- *Authenticated Key Exchange Protocols:* We presented a set of ID-based AKE protocols that can be seamlessly integrated in our proposed authentication and key exchange framework but also serve as an independent solution. The first AKE protocol is an extremely efficient and purely symmetric protocol that utilizes pairwise pairing-based keys as pre-shared secrets. We then derived more protocols by gradually adding security features, which sacrifices

some of the computational efficiency of the first protocol. The security of the protocols are based on the difficulty of solving the BDH problem and we show which security properties are achieved by each protocol. Our performance and security analysis enables the selection of the most efficient AKE protocol for particular MANET applications depending on network constraints and security needs.

Furthermore, we introduced efficient DH-based AKE protocols that establish fresh session keys by exchanging signed ephemeral DH-keys. Our protocols employ the integration step from Arazi's protocol [3] which helps to reduce computational and communication costs significantly. Our protocols are the only Arazi-type protocols that resist Nyberg & Rueppel's known-key attack [98] on Arazi's protocol while maintaining the efficiency of Arazi's original protocol and offering all additional security properties of recent Arazi protocol variants [50,119]. Our integrated protocols can be implemented using finite fields, elliptic curves or second- or third-order characteristic sequences as crypto primitives.

- *Self-organized Revocation:* We introduced a novel fully self-organized revocation scheme for IBC schemes deployed in MANETs. The revocation scheme can be seamlessly integrated into our authentication and key exchange framework as well as in any other pairing-based IBC scheme for MANETs that does not provide a mechanism for revocation, e.g. in [33, 77]. Unlike the only other revocation scheme for IBC schemes in MANETs to date [124], our scheme does not require distributed on-line KGCs in the network or any other KGC for that matter. Furthermore, our scheme is the first revocation scheme for MANETs that enables nodes to efficiently and securely revoke their own keys. Importantly, the scheme allows newly joining nodes to securely and efficiently obtain previous accusations. All communications in our revocation scheme are cryptographically protected but unlike other proposed schemes do not require digital signatures. Once pairwise pre-shared keys are computed, all messages require only the computations of symmetric cryptographic primitives such as MACs and hash functions, which makes our scheme very efficient. Security

parameters δ and ε and performance parameter m make our scheme adjustable to the hostility of the MANET environment and the degree of resource constraints of network and devices. In our extensive security analysis we have showed that our revocation scheme prevents all attacks by outsiders, individual selfish or malicious nodes as well as roaming adversaries. Furthermore, we showed how security parameters δ and ε as well as monitoring schemes with α and β can be selected to prevent attacks by colluding nodes.

- *Key Renewal:* We presented a new format for ID-based public keys that allows for key renewal at any time. Unlike other proposed key formats, e.g. [124], our format allows key renewal for the same expiry date t_x, i.e. before the start of a new expiry interval t_{x+1}. The key renewal scheme is complimentary to our revocation scheme and allows users to request new keys when their current ones have been revoked. We showed in our security analysis that Sybil and other impersonation attacks on our key renewal scheme are prevented due to the use of ID-based keys and the fact that an on-line KGC checks the identity of every node before issuing keys.

- *Key Escrow Prevention:* We were the first to discuss key escrow in the context of MANETs and we analyzed the probability of successful attacks by malicious KGCs abusing their power as key escrow. From our analysis we concluded that the special features of MANETs—such as short communication range and node mobility—significantly reduce the escrow capabilities of malicious KGCs compared to infrastructure and/or wired networks. We then introduced the novel concept of spy nodes that can be utilized by KGCs to significantly increase their escrow capabilities in MANETs. As part of our analysis we discussed countermeasures to either increase or decrease key escrow capabilities.

8.2 Future Work

The proposed security solutions for MANETs can be enhanced in several ways to further improve their performance. In addition, our theoretical security and performance analysis could be supplemented by simulation and experimental results. In this section we describe some of these possible enhancements and extensions.

- *AKE Protocols:* Our analysis of the proposed ID-based and integrated AKE protocols proved a large number of security properties and the resistance to most common attacks on AKE protocols. This could be supplemented by a formal security proof. We already outlined directions for such proofs in the Canetti and Krawczyk security model for key exchange protocols [25]. We believe that this security model is a good choice for formally proving the security of the proposed ID-based AKE protocols because the presented protocols are all designed using one of the three authenticators, which have been proven to be secure in this model; namely MAC, digital signature and public key encryption-based authenticators [10, 25].

- *Revocation Scheme:* As a next step in the performance and security evaluation of our proposed revocation scheme, the behavior of the algorithms could be further studied in simulations. For instance, security parameters δ and ε and performance parameter m could be varied in a simulation to study the security performance trade-off. In particular, the propagation delay of accusation messages in an m-hop neighborhood should be studied including the total time from observing malicious behavior to the actual revocation of a key. Furthermore, it would be of interest to analyze how monitoring schemes with different false positive and false negative rates α and β influence the scheme's performance as well as resistance to attacks by colluding nodes. Latter results could be then compared to our theoretical results. In addition, node mobility and its impact on security and performance could be simulated for different mobility patterns. As a next step, the simulated performance results of our scheme should be compared to revocation schemes that use a sign & broadcast approach. Finally, the performance of our revocation scheme for

MANETs and WMNs should be both simulated and compared to measure the performance gain by taking advantage of sporadic infrastructure access. As an extension to our security analysis of our key revocation scheme for WMNs, the detection of malicious APs or MRs that refuse forwarding recent $\mathcal{KRL}$s to clients or uploading client accusations to the RS could be studied.

- *Key Escrow:* The spy model could be further examined by simulating several mobility patterns of regular and spy nodes and analyzing the impact on the probability of successful key escrow attacks P_{suc}. Following this, simulations could consider deployment of static spy nodes at certain key locations, such as network bottlenecks, or analyze the effects of different routing protocols on P_{suc}. In addition, more applications for the usage of monitoring nodes could be explored.

- *Extensions to AKE Framework:* The presented authentication and key exchange framework can be further extended to provide additional features and functionalities. Some extensions can be easily added by utilizing keys derived from the pairwise pre-shared keys in the framework. For example, many proposed security solutions assume the pre-existence of shared keys but do not provide mechanisms for establishing such keys. These solutions could be implemented making use of the pre-shred keys. For instance, one of the existing symmetric key-based secure routing protocols could be integrated in the framework by using the pairwise pre-shared keys to secure each hop on a multi-hop routing path, e.g. [65, 63, 100].

Bibliography

[1] C. Adams, G. Kramer, S. Mister, and R. Zuccherato. On the Security of Key Derivation Functions, Information Security, LNCS 3225, Springer Verlag, pp. 134-145, 2004.

[2] American National Standard X9.63: Key Agreement and Key Transport Using Elliptic Curve Cryptography.

[3] B. Arazi. Integrating a Key Distribution Procedure into the Digital Signature Standard, *Electron. Lett.*, vol. 29, no. 11, pp. 966-967, 1993.

[4] N. Asokan and P. Ginzboorg. Key Agreement in Ad Hoc Networks, *Computer Communications*, vol. 23, no. 17, pp. 1627-1637, 2000.

[5] B. Awerbuch, R. Curtmola, D. Holmer, C. Nita-Rotaru, and H. Rubens. Mitigating Byzantine Attacks in Ad Hoc Wireless Networks, Technical Report 1, Wireless Communication Lab, Johns Hopkins University, Baltimore, Maryland, March 2004.

[6] J. Baek and Y. Zheng. Identity-Based Threshold Signature Scheme from the Bilinear Pairings, *International Conference on Information Technology: Coding and Computing (ITCC'04)*, vol. 1, pp. 124-128, 2004.

[7] D. Balfanz, D.K. Smetters, P. Stewart, and H. Chi Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks, *Proceedings of Network and Distributed System Security Symposium 2002 (NDSS '02)*, 2002.

[8] P.S.L.M. Barreto, H.Y. Kim, and M. Scott. Efficient Algorithms for Pairing-Based Cryptosystems, *Advances in Cryptology - CRYPTO 2002*, LNCS 2442, Springer Verlag, pp. 354-368, 2002.

[9] P.S.L.M. Barreto, B. Lynn, and M. Scott. Efficient Implementation of Pairing-Based Cryptosystems, *Journal of Cryptology*, vol. 17, no. 4, pp. 321-334, 2004.

[10] M. Bellare, R. Canetti, and H. Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols, *Proceedings of the 30th Annual Symposium on the Theory of Computing*, ACM, pp. 419-428, 1998. Full version available at `http://www-cse.ucsd.edu/users/mihir/papers/modular.pdf`

[11] S.M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks, *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, IEEE Computer Society, ISBN: 0-8186-2825-1, pp. 72-84, 1992.

[12] F. Bennett, D. Clarke, J.B. Evans, A. Hopper, A. Jones, and D. Leask. Piconet: Embedded Mobile Networking, *IEEE Pers. Commun.*, vol. 4, no. 5, pp. 8-15, 1997.

[13] G.M. Bertoni, L. Chen, P. Fragneto, K.A. Harrison, and G. Pelosi. Computing Tate Pairing on Smartcards. Available at `http://www.st.com/stonline/products/families/smartcard/ches2005_v4.pdf`.

[14] S. Bhargava and D.P. Agrawal. Security Enhencements in AODV Protocol for Wireless Ad Hoc Networks, *VTC 2001 Fall*, vol.4, pp. 2143-2147, 2001.

[15] Bluetooth® SIG, *Specification of the Bluetooth System*, Version 1.1; February 22, 2001. Available at `https://www.bluetooth.com`.

[16] M. Bohge and W. Trappe. An Authentication Framework for Hierarchical Ad Hoc Sensor Networks, *Proceedings of the 2003 ACM workshop on Wireless security*, ISBN: 1-58113-769-9, ACM Press, pp. 79-87, 2003.

[17] D. Boneh and M. Franklin. Identity-Based Encryption from the Weil Pairing, *Advances in Cryptology - CRYPTO '2001*, LNCS 2139, Springer Verlag, pp. 213-229, 2001.

[18] C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*, ISBN-13: 978-3540431077, Springer Verlag, 2003.

[19] C. Boyd, W. Mao, and K.G. Paterson. Key Agreemet Using Statically Keyed Authenticators, *Applied Cryptography and Network Security (ACNS) 2004*, LNCS 3089, Springer Verlag, pp. 248-262, 2004.

[20] K. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. Olsson. Detecting Disruptive Routers: A Distributed Network Monitoring Approach, *IEEE Symposium on Security and Privacy*, 1998.

[21] S. Buchegger and J. Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation of Nodes: Fairness in Dynamic Ad-hoc Networks), *MOBIHOC '02*, 2002.

[22] L. Buttyán and J.-P. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks, *Mobile Network Applications*, special issue on Mobile Ad Hoc Networks, Kluwer Academic Publishers, vol. 8, no. 5, pp. 579-592, 2003.

[23] M. Cagalj, S. Capkun, and J.P. Hubaux. Key Agreement in Peer-to-Peer Wireless Networks, *Proceedings of IEEE, Special Issue on Security and Cryptography*, vol. 94, no. 2, 2006.

[24] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. Cryptology ePrint Archive: 2003/182, 2003. Available at `http://eprint.iacr.org/2003/182`.

[25] R. Canetti and H. Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels, *Advances in Cryptology - EUROCRYPT '01*, LNCS 2045, Springer Verlag, pp. 453-474, 2001. Full version available at `http://eprint.iacr.org/2001/040`.

[26] S. Čapkun, J.-P. Hubaux, and L. Buttyán. Self-Organized Public-Key Management for Mobile Ad Hoc Networks, *IEEE Trans. Mobile Comput.*, vol. 2, no. 1, pp. 52-64, 2003.

[27] L. Chen and C. Kudla. Identity Based Authenticated Key Agreement Protocols from Pairings. *Hewlett-Packard Technical Report HPL-2003-25 20030212*, HP Laboratories, 2003.

[28] L. Chen, K. Harrison, D. Soldera, and N.P. Smart. Applications of Multiple Trust Authorities in Pairing Based Cryptosystems, *InfraSec 2002*, LNCS 2437, Springer Verlag, pp. 260-275, 2002.

[29] S. Cheung and K. Levitt. Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection, *Proceedings of New Security Pradigms Workshop*, 1997.

[30] T. Clancy and H. Tschofenig. EAP Generalized Pre-Shared Key (EAP-GPSK), Internet Draft, Work in Progress, <draft-ietf-emu-eap-gpsk-04.txt>, March 2007.

[31] J. Clulow and T. Moore. Suicide for the Common Good: a New Strategy for Credential Revocation in Self-Organized Systems, *ACM SIDOPS Operating Systems Reviews*, vol. 40, no. 3, pp. 18-21, 2006.

[32] C. Crépeau and C.R. Davis. A Certificate Revocation Scheme for Wireless Ad Hoc Networks, *Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, ACM Press, ISBN: 1-58113-783-4, pp. 54-61, 2003.

[33] H. Deng, A. Mukherjee, D.P. Agrawal. Threshold and Identity-based Key Management and Authentication for Wireless Ad Hoc Networks, *International Conference on Information Technology: Coding and Computing (ITCC'04)*, vol. 1, pp. 107-115, 2004.

[34] W. Diffie and M. Hellman. New Directions in Cryptography, *IEEE Trans. on Inform. Theory*, vol. IT-22, no. 6, pp. 644-654, 1976.

[35] J. R. Douceur. The Sybil Attack, *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, LNCS 2429, Springer Verlag, pp. 251-260, 2002.

[36] L. Eschenauer and V.D. Gligor. A Key-Management Scheme for Distributed Sensor Networks, *9th ACM conference on Computer and Communications Security*, ISBN: 1-58113-612-9, ACM Press, pp. 41-47, 2002.

[37] W. Fifer and F. Bruno. The Low-Cost Packet Radio, *Proceedings of the IEEE*, vol. 75, no. 1, pp. 33-42, 1987.

[38] FIPS 180-3, US Federal Information Processing Standard, Secure Hash Standard, February 2004.

[39] FIPS 198, US Federal Information Processing Standard, The Keyed-Hash Message Authentication Code (HMAC), March 2002.

[40] Galileo –European Satellite Navigation System, `http://ec.europa.eu/ /dgs/energy_transport/galileo`.

[41] C. Gentry. Certificate-Based Encryption and the Certificate Revocation Problem, *Advances in Cryptology- EUROCRYPT '2003*, LNCS 2656, Springer Verlag, pp. 272-293, 2003.

[42] G. Gilder. Telecosm: How Infinite Bandwidth Will Revolutionize Our World, published on KurzweilAI.net, `http://www.kurzweilai.net/`, Feb. 22, 2001.

[43] M. Girault. Self-Certified Public Keys, *Advances in Cryptology- EUROCRYPT '91*, LNCS 547, Springer Verlag, pp. 490-497, 1991.

[44] S. Gokhale and P. Dasgupta. Distributed Authentication for Peer-to-Peer Networks, *Symposium on Applications and the Internet Workshops 2003 (SAINT'03 Workshops)*, IEEE Computer Society 2003, ISBN: 0-7695-1873-7, pp. 347-353, 2003.

[45] S.W. Golomb. *Shift Register Sequences*, Revised Edition, Aegean Park Press, May 1982.

[46] G. Gong and L. Harn. Public-key Cryptosystems Based on Cubic Finite Field Extensions, *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2601-2605, 1999.

[47] G. Gong, L. Harn, and H. Wu. The GH Public-Key Cryptosystem, *Proceedings of Selected Areas in Cryptography (SAC) 2001*, LNCS 2259, Springer Verlag, pp. 284-300, 2001.

[48] Global Positioning System, `http://www.gps.gov`.

[49] Global System for Mobile Communications (GSM), `http://gsm.org`.

[50] L. Harn, M. Mehta, and Wen-Jung Hsin. Integrating Diffie-Hellman Key Exchange into the Digital Signature Algortihm (DSA), *IEEE Commun. Lett.*, vol. 8, no. 3, pp. 198-200, 2004.

[51] F. Hess. Efficient Identity Based Signature Schemes Based on Pairings. *Selected Areas in Cryptography –SAC 2002*, LNCS 2595, Springer Verlag, pp. 310-324, 2003.

[52] K. Hoeper and G. Gong. Models of Authentication in Ad Hoc Networks and Their Related Network Properties, Technical Report CACR 2004-03, Centre for Applied Cryptographic Research, Waterloo, Canada, 2004.

[53] K. Hoeper and G. Gong. Identity-Based Key Exchange Protocol for Ad Hoc Networks, *Canadian Workshop of Information Theory -CWIT '05*, pp. 127-130, 2005.

[54] K. Hoeper and G. Gong. Short Paper: Limitations of Key Escrow in Identity-Based Schemes in Ad Hoc Networks, *Security and Privacy for Emerging Areas in Communication Networks SecureComm 05*, pp. 403-405, 2005.

[55] K. Hoeper and G. Gong. Efficient Key Exchange Protocols for Wireless Networks and Mobile Devices, Technical Report CACR 2005-31, Centre for Applied Cryptographic Research, Waterloo, Canada, 2005.

[56] K. Hoeper and G. Gong. Integrated DH-like Key Exchange Protocols from LUC, GH and XTR, *IEEE International Symposium on Information Theory (ISIT) 2006*, pp. 922-926, 2006.

[57] K. Hoeper and G. Gong. Bootstrapping Security in Mobile Ad Hoc Networks Using Identity-Based Schemes with Key Revocation, Technical Report CACR 2006-04, Centre for Applied Cryptographic Research, Waterloo, Canada, January 2006.

[58] K. Hoeper and G. Gong. Pre-Authentication and Authentication Models in Ad Hoc Networks, book chapter in *Wireless Network Security*, edited by Y. Xiao, X. (Sherman) Shen, and D.-Z. Du, Springer Verlag, ISBN: 978-0-387-28040-0, 2007.

[59] K. Hoeper and G. Gong. Preventing or Utilizing Key Escrow in Identity-Based Schemes Employed in Mobile Ad Hoc, *International Journal of Security and Networks (IJSN)*, Special Issue on Cryptography in Networks, vol. 2, issue 3/4, pp. 239-250, 2007.

[60] K. Hoeper and G. Gong. Bootstrapping Security in Mobile Ad Hoc Networks Using Identity-Based Schemes, book chapter in *Security in Distributed and Networking Systems*, Computer and Network Security – vol. 1, edited by Y. Xiao and Y. Pan, World Scientific Publishing Co., ISBN: 978-981-270-807-6, to be published Fall 2007.

[61] K. Hoeper and G. Gong. Key Revocation for Identity-Based Schemes in Mobile Ad Hoc Networks, *International Conference on AD-HOC Networks & Wireless (AD HOC NOW '06)*, LNCS 4104, Springer Verlag, pp. 224-237, 2006.

[62] K. Hoeper and Lidong Chen. Where EAP Security Claims Fail, *International Conference on Heterogenous Networking for Quality, Reliability, Security and Robustness, QShine 2007*, 2007.

[63] Y.-C. Hu, D.B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing in Mobile Wireless Ad-Hoc Networks, *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '02)*, pp. 3-13, 2002.

[64] Y.-C. Hu, A. Perrig, and D.B. Johnson. Wormhole Detection in Wireless Ad Hoc Networks, Technical report, Rice University Department of Computer Science, June 2002.

[65] Y.-C. Hu, A. Perrig, and D.B. Johnson. Ariadne: a Secure On-Demand Routing Protocol for Ad Hoc Networks, *MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking*, pp. 12-23, 2002.

[66] Y.-C. Hu, A. Perrig, and D.B. Johnson. Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols, *WiSe '03: Proceedings of the 2003 ACM workshop on Wireless security*, pp. 30-40, 2003.

[67] J.P. Hubaux, L. Buttyán and S. Čapkun. The Quest for Security in Mobile Ad Hoc Networks, *ACM Symposium on Mobile Networking and Computing –MobiHOC 2001*, 2001.

[68] IEEE Standard 802.11-1999, Institute of Electrical and Electronics Engineers, "Standard for Local and metropolitan area networks - specific requirements - part 11: Wireless LAN Medium Access Control and Physical Layer specifications", 1999.

[69] IEEE Standard 802.11i, Institute of Electrical and Electronics Engineers, "Supplement to Standard for Telecommunications and Information Exchange Between Systems - LAN/MAN Specific Requirements - part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specification for Enhanced Security", July 2004.

[70] IEEE Standard 802.15.1-2005, Institute of Electrical and Electronics Engineers, "Standard for Local and metropolitan area networks - specific require-

ments - part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)", 2005.

[71] IEEE Standard 802.15.4-2003, Institute of Electrical and Electronics Engineers, "Standard for Local and metropolitan area networks - specific requirements - part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)", 2003.

[72] IEEE Standard 802.16-2004, Institute of Electrical and Electronics Engineers, "Standard for Local and metropolitan area networks - specific requirements - part 16: Air Interface for Fixed Broadband Wireless Access Systems", October 2004.

[73] IEEE Standard 802.16e, Institute of Electrical and Electronics Engineers, "Standard for Local and metropolitan area networks - Air Interface for Fixed and Mobile Broadband Wireless Access Systems", Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, Corrigendum 1, February 2006.

[74] Offical Infrared Data Association (IrDA) Homepage, `http://www.irda.org`.

[75] D.B. Johnson and D.A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks, *Mobile Computing*, Kluwer Academic Publishers, vol. 353, chapter 5, pp. 153-181, 1996.

[76] R.E. Kahn, S.A. Gronemeyer, J. Burchfiel, and R.C. Kunzelman. Advances in Packet Radio Technology, *Proceedings of the IEEE*, vol. 66, no. 11, pp. 1468-1496, 1978.

[77] A. Khalili, J. Katz, and W. Arbaugh. Toward Secure Key Distribution in Truly Ad-Hoc Networks, *2003 Symposium on Applications and the Internet Workshops (SAINT 2003)*, IEEE Computer Society, ISBN: 0-7695-1873-7, pp. 342-346, 2003.

[78] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks, *International Conference on Network Protocols (ICNP) 2001*, 2001.

[79] H. Krawczyk. SKEME: A Versatile Secure Key Exchange Mechanism for Internet, *Proceedings of the 1996 Symposium on Network and Distributed System Security (SNDSS '96)*, pp. 114-127, 1996.

[80] L. Lamport. Password Authentication with Insecure Communication, *Communication of the ACM*, vol. 24, no. 11, pp. 770-772, 1981.

[81] B. Lee, C. Boyd, E. Dawson, K. Kim, J. Yang, and S. Yoo. Secure Key Issuing in ID-based Cryptography, *CRPIT '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, Australian Computer Society, Inc., pp. 69-74, 2004.

[82] S. Lee, B. Han, and M. Shin. Robust Routing in Wireless Ad Hoc Networks, *International Conference on Parallel Processing Workshop (ICPPW'02)*, 2002.

[83] A.K. Lenstra and E.R. Verheul. The XTR Public Key System, *Advances in Cryptology - CRYPTO 2000*, LNCS 1880, Springer Verlag, pp. 1-9, 2000.

[84] D. Liu and P. Ning. Location-Based Pairwise Key Establishments for Static Sensor Networks, *1st ACM Workshop Security of Ad Hoc and Sensor Networks (SASN) '03*, ISBN: 1-58113-783-4, ACM Press, pp. 72-82, 2003.

[85] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-Securing Ad Hoc Wireless Networks, *Seventh IEEE Symposium on Computers and Communications (ISCC '02)*, 2002.

[86] J. Luo, J.-P. Hubaux, and P.Th. Eugster. DICTATE: DIstributed CerTification Authority with probabilisTic frEshness for Ad Hoc Networks, *IEEE Trans. Dependable and Secur. Comp.*, vol. 2, no. 4, pp. 311-323, 2005.

[87] B. Lynn. Authenticated Identity-Based Encryption, *Cryptology ePrint Archive*, Report 2002/072, 2002. Available at `http://eprint.iacr.org/2002/072`.

[88] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks, *Sixth Annual International Conference on Mobile Commmunication and Networking*, 2000.

[89] A.J. Menezes, P.C. von Orschot, and S.A. Vanstone. *Handbook of Applied Cryptography*, 1997 by CRC press LLC.

[90] A.J. Menezes, M. Qu, and S. Vanstone. Some New Key Agreement Protocols Providing Implicit Authentication, *Selected Areas in Cryptography –SAC '95*, pp. 22-32, 1995.

[91] T.S. Messerges, J. Cukier, T.A.M. Kevenaar, L. Puhl, R. Struik, and E. Callaway, A Security Design for a General Purpose, Self-Organizing, Multihop Ad Hoc Wireless Network, *1st ACM workshop on Security of ad hoc and sensor networks (SASN) '03*, ISBN: 1-58113-783-4, ACM Press, pp. 1-11, 2003.

[92] V.S. Miller. The Weil Pairing, and Its Efficient Calculation, *Journal of Cryptology*, vol. 17, no. 4, pp. 235-261, 2004.

[93] G.E. Moore. Cramming More Components onto Integrated Circuits, *Electronics Magazine*, vol. 38, no. 8, 1965.

[94] W.B. Mueller and W. Noebauer. Cryptanalysis of the Dickson-Scheme, *Advances in Cryptology - EUROCRYPT '85*, LNCS 219, Springer Verlag, pp. 50-61, 1986.

[95] National Institute of Standards and Technology (NIST), Digital Signature Standard (DSS), *Federal Information Processing Standards Publication*, FIPS PUB 186-2, Reaffirmed, Jan. 27, 2000.

[96] National Institute of Standards and Technology (NIST), Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, March 2006.

[97] B.C. Neumann and T. Ts'o. Kerberos: An Authentication Service for Computer Networks, *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 33-38, 1994.

[98] K. Nyberg and R.A. Rueppel. Weaknesses in Some Recent Key Agreement Protocols, *Electron. Lett.*, vol. 30, no. 1, pp. 26-27, 1994.

[99] J.H. Oh, K.K. Lee, and S.-J. Moon. How to Solve Key Escrow and Identity Revocation in Identity-Based Encryption Schemes, *First International Conference Information Systems Security (ICISS 2005)*, LNCS 3803, Springer Verlag, pp. 290-303, 2005.

[100] P. Papadimitratos and Z. Haas. Secure Routing for Mobile Ad Hoc Networks, *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, 2002.

[101] K.G. Paterson. Cryptography from Pairings: a Snapshot of Current Research, *Information Security Technical Report*, vol. 7, no. 3, pp. 41-54, 2002.

[102] RFC 2449. OpenPGP Message Format, J. Callas, L. Donnerhacke, H. Finney, and R. Thayer, November 1998.

[103] RFC 2459. Internet X.509 Public Key Infrastructure Certificate and CRL Profile, R. Housley, W. Ford, W. Polk, and D. Solo, January 1999.

[104] RFC 2560. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, June 1999.

[105] RFC 3579. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP), B. Aboba, and P. Calhoun, September 2003.

[106] RFC 3588. Diameter Base Protocol, P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, September 2003.

[107] RFC 3748. Extensible Authentication Protocol (EAP), B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Lefkowetz, June 2004.

[108] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairings, *The 2000 Symposium on Cryptography and Information Security*, 2000.

[109] A.O. Salako. Authentication in Ad hoc Networking, *In Proceedings of London Communications Symposium 2002*, 2002.

[110] K. Sanzgiri, B. Dahill, B. Levine, and E. Belding-Royer. A Secure Routing Protocol for Ad Hoc Networks, *International Conferenc on Network Protocols (ICNP)*, 2002.

[111] A. Shamir. How to Share a Secret, *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[112] A. Shamir. Identity-based Cryptosystems and Signature Schemes, *Advances in Cryptology- CRYPTO '84*, LNCS 196, Springer Verlag, pp. 47-53, 1984.

[113] Official Homepage of the SmartDust project, `http://robotics.eecs.berkeley.edu/~pister/SmartDust/`.

[114] P. Smith. A Public-Key Cryptosystem and a Digital Signature System Based on the Lucas Function Analogue to Discrete Logarithms, *Advances in Cryptology - ASIACRYPT '94*, LNCS 917, Springer Verlag, pp. 357-364, 1995.

[115] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-Hoc Wireless Networks, *In Proceedings of the 7th International Workshop on Security Protocols*, LNCS 1796, Springer Verlag, pp. 172-194, 1999.

[116] F. Stajano. The Resurrecting Duckling—What Next?, *Proceedings of the 8th International Workshop on Security Protocols*, LNCS 2133, Springer Verlag, pp. 204-214, 2000.

[117] C.E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers, *SIGCOMM '94: Proceedings of the conference on Communications architectures, protocols and applications*, ACM press, pp. 234-244, 1994.

[118] C.E. Perkins and E.M. Royer. Ad-hoc On-Demand Distance Vector Routing, *WMCSA '99: Second IEEE Workshop on Mobile Computer Systems and Applications*, IEEE Computer Society, pp. 90-100, 1999.

[119] R.C.-W. Phan. Fixing the Integrated Diffie-Hellman-DSA Key Exchange Protocol, *IEEE Commun. Lett.*, vol. 9, no. 6, pp. 570-572, 2005.

[120] A. Weimerskirch and D. Westhoff. Zero Common-Knowledge Authentication for Pervasive Networks, *Tenth Annual International Workshop on Selected Areas in Cryptography (SAC 2003)*, 2003.

[121] A. Weimerskirch and D. Westhoff, Identity Certified Authentication for Ad-hoc Networks, *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks (SASN)*, 2003, ACM Press, ISBN: 1-58113-783-4, pp. 33-40, 2003.

[122] M.G. Zapata and N. Asokan. Securing Ad hoc Routing Protocols, *Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002)*, pp. 1-10, 2002.

[123] Y. Zhang, W. Lee, and Y.-A. Huang. Intrusion Detection Technniques for Mobile Wireless Networks, *ACM J. Wireless Net.*, vol. 9, no. 5, pp.545-556, 2003.

[124] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Securing Mobile Ad Hoc Networks with Certificateless Public Keys. *IEEE Trans. Dependable Secur. Comput.*, vol. 3, no. 4, pp. 386-399, 2006.

[125] L. Zhou and Z.J. Haas. Securing Ad Hoc Networks, *IEEE Network Journal*, vol. 13, no. 6, pp. 24-30, 1999.

[126] ZigBee® Alliance, ZigBee v.1.0, 2004, available at `https://www.zigbee.org`.