Key Management in Mobile Ad Hoc Networks

Johann van der Merwe

B.Sc. Engineering (Electronic)

University of Natal, South Africa

Submitted in fulfillment of the academic requirements
for the degree of M.Sc. in Engineering
in the School of Electrical, Electronic and Computer Engineering
at the University of KwaZulu-Natal, South Africa

November 17, 2005

Opgedra aan my ma, Georgina, vir al haar liefde en ondersteuning.

This document was created in LaTeX 2$_\varepsilon$

As the candidate's supervisor I have approved this dissertation for submission.

Signed: _____

Name: Mr. Stephen A. McDonald

Date: November 17, 2005

As the candidate's co-supervisor I have approved this dissertation for submission.

Signed: _____

Name: Prof. Dawoud S. Dawoud

Date: November 17, 2005

Abstract

Mobile ad hoc networks (MANETs) eliminate the need for pre-existing infrastructure by relying on the nodes to perform all network services. The connectivity between the nodes is sporadic due to the shared, error-prone wireless medium and frequent route failures caused by node mobility. *Fully* self-organized MANETs are created solely by the end-users for a common purpose in an ad hoc fashion. Forming peer-to-peer security associations in MANETs is more challenging than in conventional networks due to the lack of central authority.

This thesis is mainly concerned with peer-to-peer key management in fully self-organized MANETs. A key management protocol's primary function is to bootstrap and maintain the security associations in the network, hence to create, distribute and revocate (symmetric or asymmetric) keying material as needed by the network security services. The *fully* self-organized feature means that the key management protocol cannot rely on any form of off-line or on-line trusted third party (TTP).

The first part of the thesis gives an introduction to MANETs and highlights MANETs' main characteristics and applications. The thesis follows with an overall perspective on the security issues in MANETs and motivates the importance of solving the key management problem in MANETs.

The second part gives a comprehensive survey on the existing key management protocols in MANETs. The protocols are subdivided into groups based on their main characteristic or design strategy. Discussion and comments are provided on the strategy of each group. The discussions give insight into the state of the art and show researchers the way forward.

The third part of the thesis proposes a novel peer-to-peer key management scheme for fully self-organized MANETs, called Self-Organized Peer-to-Peer Key Management (Self-OrgPKM). The scheme has low implementation complexity and provides self-organized mechanisms for certificate dissemination and key renewal without the need for any form of off-line or on-line authority.

The fully distributed scheme is superior in communication and computational overhead with respect to its counterparts. All nodes send and receive the same number of messages and complete the same amount of computation. SelfOrgPKM therefore preserves the symmetric relationship between the nodes. Each node is its own authority domain which provides an adversary with no convenient point of attack.

SelfOrgPKM solves the classical routing-security interdependency problem and mitigates impersonation attacks by providing a strong one-to-one binding between a user's certificate information and public key. The proposed scheme uses a novel certificate exchange mechanism that exploits user mobility but does not rely on mobility in anyway. The proposed certificate exchange mechanism is ideally suited for bootstraping the routing security. It enables nodes to setup security associations on the network layer in a localized fashion without any noticeable time delay.

The thesis also introduces two generic cryptographic building blocks as the basis of SelfOrg-PKM: 1) A variant on the ElGamal type signature scheme developed from the generalized ElGamal signature scheme introduced by Horster et al. The modified scheme is one of the most efficient ElGamal variants, outperforming most other variants; and 2) A subordinate public key generation scheme.

The thesis introduces the novel notion of subordinate public keys, which allows the users of SelfOrgPKM to perform self-organized, self-certificate revocation without changing their network identifiers/addresses. Subordinate public keys therefore eliminate the main weakness of previous efforts to solve the address ownership problem in Mobile IPv6. Furthermore, the main weakness of previous efforts to break the routing-security interdependence cycle in MANETs is also eliminated by a subordinate public key mechanism.

The presented ElGamal signature variant is proved secure in the Random Oracle and Generic Security Model (ROM+GM) without making any unrealistic assumptions. It is shown how the strong security of the signature scheme supports the security of the proposed subordinate key generation scheme. Based on the secure signature scheme a security argument for SelfOrgPKM is provided with respect to a general, active insider adversary model.

The only operation of SelfOrgPKM affecting the network is the pairwise exchange of certificates. The cryptographic correctness, low implementation complexity and effectiveness of SelfOrgPKM were verified though extensive simulations using ns-2 and OpenSSL. Thorough analysis of the simulation results shows that the localized certificate exchange mechanism on the network layer has negligible impact on network performance. The simulation results also correlate with efficiency analysis of SelfOrgPKM in an ideal network setting, hence assuming guaranteed connectivity. The simulation results furthermore demonstrate that network layer certificate exchanges can be triggered without extending routing protocol control packets.

Preface

The research work presented in this dissertation was performed by Johann van der Merwe, under the supervision of Mr. Stephen A. McDonald and Prof. Dawoud S. Dawoud, in the School of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal. This work was supported by ARMSCOR, the Armaments Corporation of South Africa.

Publications from this work include:

J. van der Merwe, D. Dawoud, and S. McDonald, Trustworthy Key Management for Mobile Ad Hoc Networks, in proc. Southern African Telecommunication Networks and Applications Conference (SATNAC 04), September, 6-8, 2004.

J. van der Merwe, D. Dawoud, and S. McDonald, A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks, ACM Computing Surveys (CSUR), 2004, in submission.

J. van der Merwe, D. Dawoud, and S. McDonald, Self-Organized Peer-to-Peer Key Management for Mobile Ad Hoc Networks, Southern African Telecommunication Networks and Applications Conference (SATNAC 05), 2005.

J. van der Merwe, D. Dawoud, and S. McDonald, A Survey on Peer-to-Peer Key Management for Military Type Mobile Ad Hoc Networks, in proc. Military Information and Communications Syposium of South Africa (MICSSA'05), 2005.

J. van der Merwe, D. Dawoud, and S. McDonald, Fully Self-Organized Peer-to-Peer Key Management for Mobile Ad Hoc Networks, ACM Workshop on Wireless Security (WiSe'05), Cologne, Germany, September, 2, 2005.

J. van der Merwe, D. Dawoud, and S. McDonald, A Public Key Management Scheme and Threshold-Multisignature Scheme for Mobile Ad Hoc Networks, SAIEE Transactions, 2005, to appear.

The entire dissertation, unless otherwise indicated, is the student's own original work and has not been submitted in part, or in whole, to any other university for degree purposes.

Signed: _____

Name: Johann van der Merwe

Date: November 17, 2005

Acknowledgments

I would like to thank my supervisor, Mr. Stephen McDonald. He gave me a life altering opportunity when he took me on as his MSc student. I have learnt a great deal from him on a personal and technical level. I am truly thankful for the trust he placed in me and the freedom he gave me in my work. From my undergraduate days he has always supported me and believed in my potential. I wish him only the best for the future.

It was purely by chance that Professor Dawoud S. Dawoud was assigned as my co-supervisor. This was a blessing and certainly the start of an interesting journey. I have tried to source has much of his knowledge and experience as possible. He also gave me room to develop my research abilities and trusted my ideas. I highly appreciated his guidance.

Sincere thanks to ARMSCOR, the Armaments Corporation of South Africa, who financed this research project. To Franzette Vorster and Peter Handley, thank you for your trust and support.

Ek wil ook graag baie dankie sê aan my familie, my ma, Georgina, suster, Jana en broer, Richart. Hulle liefde en ondersteuning ken geen perke nie. Ek is besonders lief vir hulle. Laastens sê ek dankie aan Roxane vir haar geduld, aanmoediging and liefde. Haar vriendskap lê na aan my hart. Geen woorde kan beskryf hoe lief ek vir haar is nie.

Contents

List of Figures

Chapter 1

Introduction

As a result of significant advances in mobile computing and wireless communication technology, mobile devices have gained sufficient communication, computational and memory resources to be interconnected. By definition mobile ad hoc networks (MANETs) differentiate themselves from existing networks by the fact that they rely on no fixed infrastructure [1]: the network has no base stations, access points, remote servers etc. All network functions are performed by the nodes forming the network; each node performs the functionality of host and router, relaying data to establish connectivity between source and destination nodes not directly within each other's transmission range.

MANETs are autonomous, multi-hop networks interconnected via wireless links. The word *ad hoc* (translated as *for this only* from Latin) implies that the network is formed in a spontaneous manner to meet an immediate demand and specific goal. *Fully* self-organized MANETs are created solely by the end-users for a common purpose in an *ad hoc* fashion. Fully self-organized MANETs can be informally visualized as a group of strangers, people who have never met before, coming together for a common purpose. These people have no prior relationships and share no common keying material on their nodes. Users therefore have to establish security associations between themselves, after network formation, without the aid of a common off-line trusted third party (TTP). Furthermore, once the network is operational, there is no form of on-line TTP to perform any key management

services. The network is therefore operated and managed by the nodes themselves, which makes MANETs dependent upon the co-operative and trusting nature of nodes [7]. The autonomous nature of mobile ad hoc networks or alternatively stated, the requirement that the network is operated primarely by the end-users, is generally referred to as *self-organization* [3] [5]. This feature makes ad hoc networks financially viable since there is no cost involved in setting up or maintaining a network architecture.

Mobile ad hoc networks have a *dynamic* network topology. Since nodes in the networks are mobile, with unrestricted movement, the configuration of the network topology can change very rapidly. The position of the nodes relative to each other may exhibit a randomly changing nature and therefore be considered as unpredictable. The lack of infrastructure, dynamic network topology and error-prone wireless connectivity result in frequent link breakages implying sporadic connectivity. Protocols for MANETs therefore need to mitigate the unreliability of basic network services by taking on a fully distributed, self-organizing nature. From a security perspective, distributing the functionality of network services to as many nodes as possible, avoids a single point of attack.

Considering the unique features of ad hoc networks, it is expected that the mechanisms proposed to guarantee the security of conventional wireline networks are not necessarily suitable or adaptable to MANETs. Special mechanisms and protocols designed specifically for ad hoc networks are necessary.

1.1 Scope of Research

This work focuses on peer-to-peer or pairwise key management in *pure* or *fully self-organized* mobile ad hoc networks. As mentioned above, by definition, fully self-organized MANETs do not rely on any form of TTP to set up security associations [3] [5]. A fully self-organized MANET is an "open" network: any user with the appropriate equipment (and software) can join and leave at random; there is no form of access control. Such a network will therefore not find application in, for example, hostile military environments, but rather in commercial, community-based environments.

1.2 Outline of Thesis

The thesis is organized as follows:

1.2.1 Chapter-2

In Chapter-2, the characteristics of mobile ad hoc networks are discussed from a security perspective. Some MANET applications in commercial and military settings are identified. After discussing the main security goals, the general adversary model assumed throughout the thesis is detailed. To illustrate the capability of adversaries in an ad hoc setting a selection of the current attacks against MANETs is discussed. Chapter-2 is ended by giving an overview of the issues in the MANET security field. The purpose of key management is identified from these security areas, thereby motivating the significance of this research.

1.2.2 Chapter-3

In Chapter-3, the existing literature on key management for MANETs is investigated. Structure is provided to the discussion by dividing the existing schemes into groups, based on the schemes' main characteristics. Each group is introduced by analysing at least the original protocol that inspired the subset. This approach allows the reader to form a global persective on the existing pairwise key management solutions for MANETs.

1.2.3 Chapter-4

Chapter-4 is dedicated to the main contribution of this thesis. A new peer-to-peer key management scheme called, Self-organized Peer-to-Peer Key Managemant (SelfOrgPKM), is proposed. SelfOrgPKM is specifically designed for *fully* self-organized MANETs, i.e. MANETs that do not rely on any form of trusted authority. As building blocks for SelfOrgPKM the notion of *subordinate public keys* and a variant on the ElGamal signature scheme is presented. These building blocks may find application in various other security schemes.

1.2.4 Chapter-5

In Chapter-5, a strong security argument for SelfOrgPKM is presented. Secondly, the communication and computational cost of SelfOrgPKM are analyzed in an ideal setting. The effectiveness of SelfOrgPKM was verified through simulation. Chapter-5 presents the simulation results in detail. Lastly, the impact of SelfOrgPKM's certificate exchange mechanism on network performance is evaluated.

1.2.5 Chapter-6

In Chapter-6 the discussions are concluded, the thesis' main contributions are summarized and future directions of research are provided.

Chapter 2

Mobile Ad Hoc Networks

2.1 Characteristics of Mobile Ad Hoc Networks

It is important to acknowledge the properties or characteristics of mobile ad hoc networks (MANETs), since these properties have a significant impact on the design of security protocols for MANETs. Although these properties are detailed in various papers [8] [3] [9] [10], security protocols that do not suit these characteristics are frequently published. The thesis is therefore started by defining the characteristics of MANETs. The security implications of the characteristics are discussed where applicable.

2.1.1 Network Infrastructure

There is no fixed or pre-existing infrastructure in an ad hoc network: all network functionality (routing, security, network management etc.) is performed by the nodes themselves. Due to the nodes' limited transmission range, data dissemination is achieved in a multi-hop fashion; nodes can therefore be considered as hosts and routers. Although the lack of infrastructure opens a new window of opportunity for attacks, the author believes the lack of infrastructure can help to ensure the survivability of the network in a very hostile environment. This holds true not only from a network security perspective, but also when

the users of the network are under physical attack (see Section-2.2.1).

Ad hoc networks may be spontaneously formed with no *a priori* knowledge of the physical location and networking environment. MANETs' lack of infrastructure thus makes it suitable for various applications where conventional networks fall short (see Section-2.2).

Some researchers have already addressed security issues in *hybrid* ad hoc networks (for example [11]). Hybrid ad hoc networks combine conventional network infrastructure with multi-hopping. This derivative of ad hoc networks will find useful application where fixed infrastructure can be extended through multi-hop networks or where the functionality (and performance) of multi-hop networks can be enhanced by relying on some infrastructure.

2.1.2 Network Topology

Nodes in ad hoc networks may be mobile resulting in a dynamic, weakly connected topology. Since node mobility is unrestricted, the topology may be unpredictable. The network will however demonstrate global mobility patterns which may not be completely random [5].

The topology is weakly connected due to transient, error-prone wireless connectivity. The users may therefore experience unavailability of essential security services. Node mobility and wireless connectivity allow nodes to spontaneously join and leave the network, which makes the network amorphous. Security services must be able to scale seamlessly with rapid changes in network density.

2.1.3 Self-Organization

MANETs cannot rely on any form of central administration or control; this is essential to avoid a single point of attack [1]. A *self-organized* MANET cannot rely on any form of *off-line* trusted third party (TTP); the network can thus be initialized by a distributed *on-line* TTP. A *pure* or *fully self-organized* MANET does not rely on anyform of TTP whatsoever [3] [5], i.e. the on-line TTP is also eliminated. Nodes will therefore only have

compatible devices with the same software installed. In the extreme case, the nodes will not even share a common set of security system parameters. The lack of a TTP may force the end-users to actively participate in the setup of security associations. A (fully) self-organized MANET has some inherent security implications:

- *Fully* self-organized MANETs are "open" in nature: similar to the internet, any user can join the network at random. Access control to *applications* will have to be provided at the application layer with a varying degree of user interaction.
- Each user will be its own authority domain, hence responsible for generating and distributing its own keying material. As pointed out by Douceur [12], any node can generate more than one identity when there is no off-line TTP. It is thus clear that it will be very difficult (if not impossible) to limit users to one and only one unique identity in a (fully) self-organized setting.
- The network will always be vulnerable to the active insider adversary (see Section-2.3.2). In fact, the Dolev-Yao adversary model [13] is too restrictive [14], for example, it fails to capture information an adversary may gain from detailed knowledge of the protocols in use. An interesting topic for future research will be the adversary model in "open" ad hoc networks.
- It will be difficult to hold malicious nodes accountable for their actions, since they can always rejoin the network under a different (new) identity.

2.1.4 Limited Resources

Nodes have limited computational, memory and energy resources in contrast to their wired predecessors. Nodes are small hand-held devices (possibly "off-the-shelf" consumer electronics) that do not hinder user mobility. In an attempt to keep the cost of these devices low, they are normally powered by a small CPU, accompanied by limited memory resources. As the devices are mobile they are battery operated. This often results in short *on* times and the possibility of power failure due to battery exhaustion, perhaps during execution of a network related function.

Devices may have limited bandwidth and transmission ranges. If it is assumed that advances in integrated circuit (IC) technology will keep on following Moore's law, computational and memory limitations will be alleviated in a matter of time. Bandwidth and transmission range (in the case of communication via radio transmissions) are unlikely to improve dramatically with respect to power consumption as both are dependent on Shannon's law and thus limited [15]. In order to achieve a higher bandwidth, a higher signal-to-noise ratio (SNR) is required which in turn requires higher transmission power [15]. Higher transmission power significantly depletes battery power, which is unlikely to improve significantly given the current rate of advancement in battery technology [16].

A security protocol that fails to optimize node and network resources will simply not be adopted in practice.

2.1.5 Poor Physical Security

Nodes are mobile and therefore cannot be locked up in a secure room or closet. These small hand-held devices are easily compromised by either being lost or stolen. It is therefore highly probable than an adversary can physically compromise one or more nodes and perform any number of tests and analysis. The adversary can also use the nodes to attack distributed network services, such as a distributed on-line certificate authority [1]. Poor physical security is not as relevant in "open" MANETs: the adversaries do not have to physically capture nodes to become an insider or to perform analysis on the protocols. The poor physical security may result in serious problems in "closed", military type MANETs where physically compromised nodes can be used to launch active, insider attacks on the network.

2.1.6 Shared Physical Medium

The wireless communication medium is accessible to any entity with the appropriate equipment and adequate resources. Accordingly, access to the channel cannot be restricted. Adversaries are therefore able to eavesdrop on communication and inject bogus messages

into the network without limitation. The shared channel and the nodes' poor physical security again emphasize that security mechanisms must be able to deal with the worst case *active*, *insider* adversary.

2.1.7 Distributed System

Considering the above properties, nodes in ad hoc networks have a symmetric relationship. This implies that they are all equal and therefore should equally distribute all of the responsibilities of providing network functions. This is not only for security reasons but to ensure reliable, available network services that places the same burden on the computational, memory and energy resources of all network participants [1] [3]. It is anticipated that a fair distributions of services will also help to alleviate selfishness [7].

2.2 Applications of Mobile Ad Hoc Networks

To understand the nature of MANETs and the origin of their unique characteristics, the potential applications of ad hoc networks are briefly considered.

Ad hoc networks have applications in two major fields: military and commercial environments.

2.2.1 Military Applications

The origin of networks that rely on no pre-existing infrastructure can be traced back to the early 1970s with the DARPA and PRNET projects [17] [10], where the initial focus was on military applications.

The application of ad hoc networks in a military environment is particularly attractive because of their lack of infrastructure and self-organizing nature. Consider conventional networks that rely on infrastructure such as base stations: the infrastructure introduces points of vulnerability which may be attacked and, if eliminated, dismantle the operation

of the entire network. In battlefield scenarios robust and guaranteed communication is essential with potentially fatal consequences if compromised. Ad hoc networks can continue to exist even in the event of nodes disappearing or becoming disconnected due to poor wireless connectivity, moving out of range, physical attack on users, broken nodes, battery depletion or physical node damage.

Applications such as sensor networks [18], positional communication systems [19] [20] and tactical ad hoc networks [21] will continue to be one of the driving forces behind ad hoc network development.

2.2.2 Commercial Applications

Commercial applications of ad hoc networks may include deployment of connectivity in terrains where conventional networks, such as cellular networks, are not financially viable, cannot provide sufficient coverage or need by-passing.

Private networks or personal area networks (for the purpose of teleconferencing, video conferencing, peer-to-peer communications, ad hoc meetings, or more generally, collaborative applications of all kinds) are possible applications of ad hoc networks. It is anticipated that these applications will gain momentum as soon as the flexibility and convenience of self-organized ad hoc networking is fully appreciated and protocols are implemented with commercially available products. Take for example cellular networks, what was once seen as an impractical technology has now become a necessity.

Emergency situations caused by geopolitical instability, natural or man-made disaster could result in existing networking infrastructure being damaged or unreliable. For example, Hurricane Katrina struck New Orleans, Louisiana on August 29, 2005. The storm destroyed most of the fixed communication infrastructure as it blanketed approximately 90,000 square miles of the Unites States, a region almost as large as the United Kingdom [22]. In order to launch an effective disaster relief operation, communication is of the essence, even between a localized group of relief workers. “Open” MANETs will make it possible for relief workers from various countries to establish communication on the

fly, therefore eliminating the time penalty in setting up and managing conventional, fixed infrastructure networks. Search and rescue missions could also be conducted in locations not allowing access to existing communication networks.

Vehicular ad hoc networks allow vehicles travelling along a highway to exchange data for traffic congestion monitoring, inter vehicle communication and early warning of potential dangers ahead such as an accident, road obstruction or stationary vehicle. Several research projects have been initiated to deal with vehicular ad hoc networking [23] [24] [25].

2.3 Security Issues in Mobile Ad Hoc Networks

Security is a fundamental issue that needs resolution before ad hoc networks will experience large scale deployment [1]. Vehicular ad hoc networking is a good example of a MANET application with some serious security implications [25]: failure of the security mechanisms may result in the loss of human life. The characteristics of mobile ad hoc networks, as given in Section-2.1, pose numerous challenges in achieving conventional security goals. Since the nodes are responsible for basic network functions, like packet forwarding and routing, network operations can be easily jeopardized if countermeasures are not integrated into these network functions at the early stages of design. For example, some existing routing protocols ([26] [27] [28] [29] [30] [31] [32] [33] [34] [35]) for mobile ad hoc networks may be able to manage the dynamic network topology of mobile ad hoc networks, but none of these protocols incorporate mechanisms to prevent, tolerate or defend against attacks from malicious adversaries. Due to the close relationship between security and the characteristics of ad hoc networks these protocols will have to be fundamentally altered or re-designed to effectively incorporate security mechanisms. This is clear if one considers the number of *secure* routing protocols (Ariadne [36], SEAD [37], ARAN [38], SRP [39] SAODV [40]) that followed the insecure routing protocols.

2.3.1 Security Goals

The security objectives for ad hoc networks are similar to those of conventional networks. Ad hoc network security should consider the following fundamental attributes (adapted from [41] [1] [42]):

2.3.1.1 Confidentiality

Confidentiality services protect information from being disclosed to unauthorized entities. These services prevent entities from using the information for personal gain or malicious intent.

2.3.1.2 Data Integrity

A guarantee of uncorrupted message exchange must be provided. The data integrity service should prevent or enable detection of data modification by both authorized and unauthorized entities.

2.3.1.3 Authentication

An authentication service allows a node to verify the identity of the peer node with which it is communicating. Authentication is not only applicable for verification of the peer node's identity but also the content of the message exchanged. These attributes are known as *entity authentication* and *data origin authentication* respectively, where the latter implicitly incorporates *data integrity*.

2.3.1.4 Non-repudiation

Non-repudiation services ensure that an entity cannot deny previous commitments or actions. It binds an entity to messages, received and sent, which may infer participation in a specific protocol.

2.3.2 Adversary Model

Threats to ad hoc networks or attacks on ad hoc networks can be launched by malicious nodes as an outsider or insider with respect to the network. A clear distinction should be made between outsider and insider adversaries, since each may use different tactics to circumvent security mechanisms and should be dealt with accordingly. Any security protocol can only "guarantee" defence against an adversary that has been captured by its adversary model.

2.3.2.1 Outsider Adversaries

Insider adversaries are the most common adversaries; nodes that do not belong to the network with the intent of harming the network operation. Since an outsider does not have access to important network information, the attacks are normally based on the exploitation of the characteristics of the network or weaknesses in the underlying protocols.

2.3.2.2 Insider Adversaries

Insider adversaries are significantly more difficult to defend against than outsider adversaries. These are malicious authorized nodes performing unauthorized actions. With a strong non-repudiation service, an advantage of defending against insider adversaries in contrast to outsider adversaries, is that the insider (malicious user) is more easily identified if detected and held accountable for malicious or unauthorized actions. On the other hand, an insider has access to additional information (for example keying material, routing information) and is possibly trusted by other network nodes, making it easier for the node to breach confidentiality, corrupt fundamental network function and manipulate the content of messages without detection.

In "open" or *fully* self-organized MANETs all malicious nodes can be regarded as insider adversaries.

2.3.2.3 Passive and Active Adversaries

Both outsider and insider adversaries can be classified as being either an active or passive adversary. A *passive* adversary is only able to listen to (eavesdrop on) network traffic without disrupting protocol operation and uses the information gained to breach network security. An *active* adversary, in addition, has full control over the communication channel, making it possible for the adversary to record and inject modified or selected data into the channel. This is normally done by a compromised authorized node or an adversary impersonating an authorized node participating in network operations, which implies that insider adversaries are normally characterized as active adversaries.

2.3.3 Attacks on Mobile Ad Hoc Networks

The unique characteristics of MANETs give an adversary the opportunity to launch numerous attacks against ad hoc networks. When designing protocols for MANETs it is important to be aware of existing attacks. In "open" or *fully* self-organized MANETs, an adversary may take on as many identities as needed, the number being limited only by its available resources [12]. It is anticipated that future research will identify new attacks on "open" MANETs or at least nullify defence mechanisms against existing attacks. For example, holding nodes accountable for their actions is temporary in "open" MANETs: if the node is excluded due to misbehavior the attacker can rejoin the network with a new identity.

2.3.3.1 Eavesdropping/Traffic analysis

Eavesdropping is the most common attack where a passive adversary listens in on the communication. This attack can be successfully implemented in most networks, however in wireless networks it becomes easier as the adversary does not have to physically connect to the medium, but can monitor the traffic between communication entities from a "safe" distance. Since the adversary in general (and particularly in the case of commercial networks) can obtain the necessary communication equipment to demodulate the signal,

encryption remains the most effective solution to defend against eavesdropping.

2.3.3.2 Denial of Service

In denial of service (DoS) attacks the adversary prevents or prohibits the normal use or management of network facilities or functionality. DoS attacks can be launched at any layer of an ad hoc network to exhaust node resources [1]. On the physical and medium access control layers a DoS attack, for example *jamming*, can be employed to impair network availability. Jamming occurs when a malicious node deliberately emanates a signal from a wireless device in order to overwhelm legitimate wireless signals [43]. On the network layer an adversary could disrupt the routing fabric by manipulating routing control packets. On higher layers an adversary could attack essential services, such as the key management protocol. DoS can also be caused by the constant request of network services by an adversary - even if these services are denied. For example, *hello flooding*, can be seen as a type of DoS attack. An attacker can use *traffic diversion* to cause large amounts of data to be sent to a victim node from some set of nodes controlled by the attacker [44].

2.3.3.3 Protocol Attacks

Most security holes are discovered in protocols themselves although these protocols use secure building blocks to provide their required functionality [41] [42]. An example of this is insecure protocols without proposer authentication mechanisms, which can be attacked from both active insider and outsider adversaries (see Section-2.3.2).

2.3.3.4 Impersonation

Without strong authentication mechanisms in place, insider or outside adversaries can masquerade as an authorized user and attempt to access network services. This problem becomes even more difficult to solve in "open", fully self-organized MANETs which do not have any form of access control. In Mobile IPv6 a subset of this problem is referred

to as the address ownership problem [45].

2.3.3.5 Cryptanalytic Attacks

Cryptanalysis is the study of mathematical techniques and procedures used to *break* cryptographic techniques, procedures and algorithms, i.e. defeating the objective of the intended service [41]. Adversaries exploit cryptanalytic techniques to discover cryptographic keys or derive confidential information from the recorded encrypted data sent between communication entities. These types of attacks are however seldomly used due to cryptanalytic complexity and/or vast computational and memory requirements. The probability of a security hole existing elsewhere is generally much higher depending on what cryptographic algorithms are used. A careful analysis is still essential when selecting cryptographic algorithms to avoid a cryptanalytic attack. Factors to consider when choosing a cryptographic scheme are: is the algorithm widely used? How important is the information it is protecting? And does it have a good history of cryptanalysis? Successful cryptanalysis can also be made more difficult by limiting the amount of information available for cryptanalysis. For example, ensuring *key freshness* by independently changing encryption keys between communication sessions may significantly reduce the possibility for attack [46].

2.3.3.6 Wormhole Attacks

The wormhole attack is a severe attack against the routing protocols of ad hoc networks made possible by the shared wireless medium. In a wormhole attack, an adversary records packets (routing information) at one location in the network, tunnels the recorded data via a private network or connection to another location in the network, and re-transmits the recorded data into the network [47] [48]. This disrupts the routing infrastructure and consequently prevents a routing protocol, without a protection mechanism, from finding routes more than one or two hops from the node requesting routing services.

2.3.3.7 Sinkhole/Blackhole Attacks

During a blackhole attack the attacker forms part of the network and interferes in the route discovery phase of the routing protocol [36]. By sending forged routing packets the attacker routes all packets for another destination to itself. The attacker then disrupts the routing protocol by creating a routing *blackhole* or *sinkhole* in which all packets are dropped. The attacker could also selectively drop packets by creating a *grayhole*, for example, by forwarding all routing packets but no data packets [36].

2.3.3.8 Stealth Attacks

Stealth attacks, introduced in [44], are classified into two classes. The first class of attacks attempts to "hi-jack" or perform traffic analysis on filtered traffic to and from victim nodes. These attack are mounted, for example, by the modification of routing information. An attacker can divert traffic by using authentic routing messages to fool honest nodes into disrupting their routing tables.

The second class partitions the network and reduces goodput by disconnecting victim nodes in several ways. For example, the attacker can route a large amount of data through the victim node. This may totally consume the node's energy resources or create a perception of unavailability due the large quantities of messages being dropped by the victim. Consequently the node under attack will not be used by neighboring routers and becomes isolated.

The methods are referred to as *stealth* attacks since they minimize the cost of launching the attacks and reduce the visibility of the attacker.

2.3.3.9 Sybil Attacks

If a strong one-to-one binding between the identity and physical entities does not exist in a communication system (vouched for by an off-line trusted authority), it is always possible for an unfamiliar entity to adopt more than one identity [12]. In a *Sybil attack*, a single

inconsistent or malicious entity presents multiple identities and therefore could potentially control a substantial fraction of the system.

2.3.3.10 Rushing Attacks

A rushing attack results in a denial of service (DoS) when used against an on-demand routing protocol [49]. In on-demand routing protocols a node initiates a *route discovery* by flooding the network with a *route request*. To limit the overhead, nodes in the network forward only *one* route request originating from a specific route discovery. Assume node A issues a route request to establish a route to node B. If an attacker is *first* to forward the *route request* to the neighboring nodes of node B then the route discovering phase will always render a route of which the attacker is one hop. The attacker can thus prevent the routing protocol from finding any useful routes to node B by continuously rushing the route request to the neighbors of node B. The rushing attack can also be used against an on-demand routing protocol that predictably forwards *any* routing request for a given route discovery in contrast to a protocol in which node forwards only one route request [49].

2.3.3.11 Replay Attacks

In a replay attack on the routing infrastructure an adversary sends old route advertisements to a node causing it to update its routing table with stale routes [39]. The attacker may also record packets sent between legitimate network nodes on the application layer and replay these recorded packets in order to gain unauthorized access to network services or overload a node with outdated information resulting in denial of service.

2.3.3.12 Sleep Deprivation Torture

An adversary may interact with a node in a "legitimate" way with the purpose of consuming its battery power (energy resources). In ad hoc networks and particularly in sensor networks [18], energy resources are severely limited (Section-2.1). In order to conserve power these nodes go into sleep mode in which the channel is only periodically scanned

for signals. During a sleep deprivation torture attack, nodes are prevented from going into sleep mode until the battery of the target device is fully depleted, leaving the node disabled [50] [51].

2.3.3.13 Selfishness

Selfishness is an ad hoc network specific attack, falling under the denial of service category. In an attempt to save computational, memory and energy resources, users may become *selfish*. Mechanisms are therefore needed to enforce *co-operation* or to provide some incentive for nodes willing to participate in network operations. The impact of selfish behavior in ad hoc networks has been investigated and proven to seriously impair the network's capability to provide essential services such as routing [52] [7]. Various proposals have already been made in literature to defend against selfish nodes [52] [53] [7] [54] [55] [56] [57] [58] [59].

2.3.4 Open Problems in Mobile Ad Hoc Networking Security

Despite the evolution of MANETs over the past decade there are still a number of security related problems that are open. This means that although solutions have been proposed none seem to satisfy all of the constraints of MANETs. Figure-2.1 illustrates the areas investigated within the MANET field with particular focus on security issues. Note that this list highlights the main areas of ad hoc network security and could be expanded.

As illustrated in Figure-2.1, research in the MANET security field is concerned with a variety of different aspects.

Researchers in the ad hoc network security field initially focused on secure routing protocols [1]. The focus of these protocols (Ariadne [36], SEAD [37], ARAN [38], SRP [39], SAODV [40]) is two-fold:

1. To provide a robust routing mechanism against the dynamic topology of MANETs.
2. To provide a robust routing mechanism against malicious nodes.

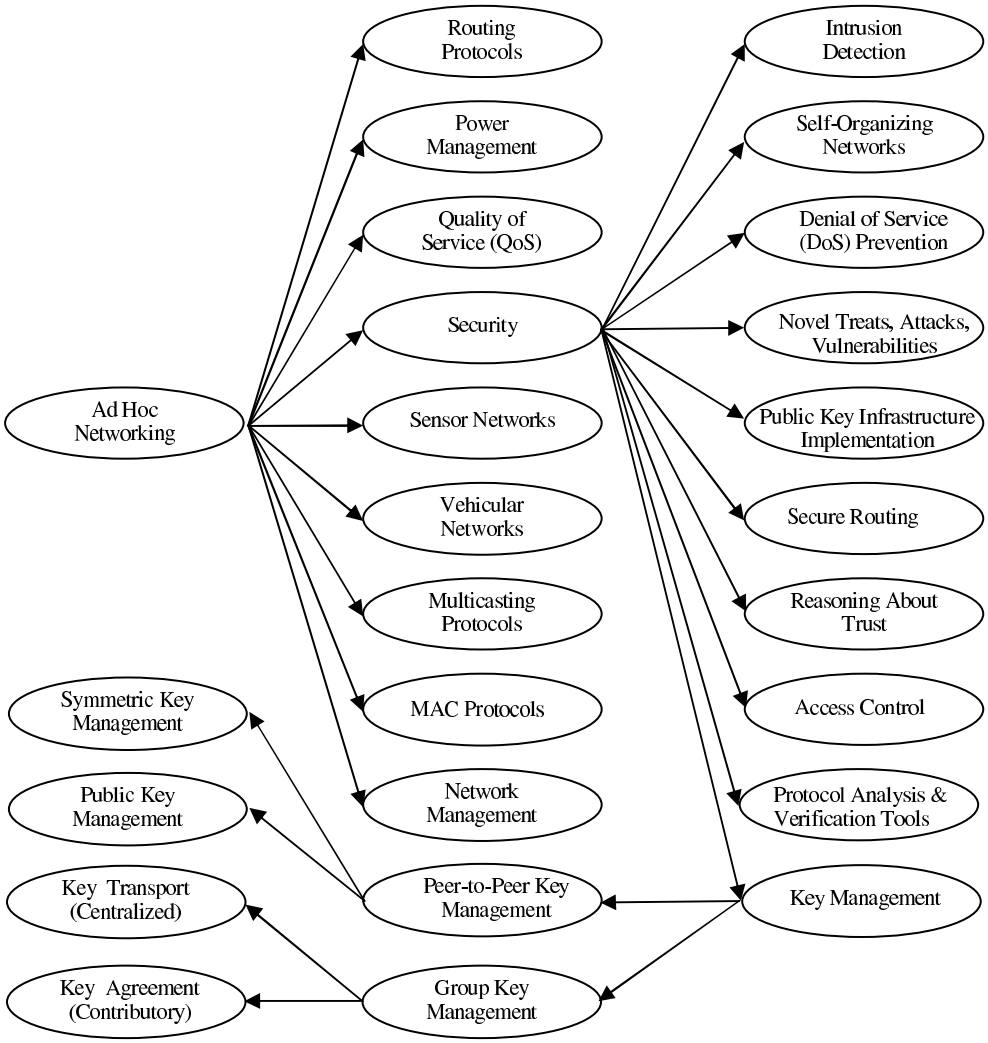

Figure 2.1: Areas of interest in ad hoc networks

Routing protocols use various security mechanisms to ensure robustness of the routing scheme. Some of these mechanisms are listed below:

1. Redundancy exploitation.
2. Diversity coding.
3. Authenticated route discovery and network nodes.
4. Guaranteed route discovery.
5. Route maintenance techniques.
6. Fault or intrusion tolerant mechanisms.
7. Cryptographic techniques, procedures, schemes, tools or mechanism.

For example, routing schemes may exploit redundancy by establishing multiple routes from source to destination (as easily achieved by ZRP [31], DSR [27], TORA [29], AODV [33]) [1]. By sending data via all these routes, the redundancy will ensure that all data arrives at the destination. An alternative mechanism to sending data via redundant routes is *diversity coding* [60]. Diversity coding takes advantage of redundant routes in a more bandwidth efficient way by not re-transmitting the messages. Rather, it transmits limited redundant information through additional routes for the purpose of error detection and correction.

All of these mechanisms have various degrees of effectiveness. It is widely acknowledged that cryptographic techniques can provide some of the strongest mechanisms to ensure the authenticity, integrity and confidentiality of routing information.

Secure key management with a high availability feature is at the center of providing network security [41]. However, all routing schemes neglect the crucial task of secure key management and assume pre-existence and pre-sharing of secret and/or private/public key pairs [1]. This leaves key management considerations as an open research area in the ad hoc network security field.

Chapter 3

A Survey on Peer-to-Peer Key Management for Mobile Ad Hoc Networks

3.1 Introduction

This chapter reviews the most popular peer-to-peer key management protocols proposed to date for mobile ad hoc networks (MANETs). The protocols are sub-divided into groups based on their design strategy or main characteristic. The existing schemes are discussed and comments are provided within the context of the assumptions made by their authors; the primary differentiation between the existing schemes can be made based on the assumption of an off-line and/or on-line trusted authority. Although the main contribution of this thesis is a peer-to-peer key management scheme for fully self-organized MANETs, the discussions on the existing schemes are also applicable to schemes that assume the existence of an off-line and/or on-line trusted third party.

Before introducing the different protocol groups, it is necessary to clarify what is meant by key management. The subsequent section also provides definitions and terminology for the different properties and requirements of key management schemes.

3.1.1 Defining Key Management

A *keying relationship* is the state wherein network nodes share keying material for use in cryptographic mechanisms [41]. The keying material can include public/private key pairs, secret keys, initialization parameters and non-secret parameters supporting key management in various instances. Key management can be defined as a set of techniques and procedures supporting the establishment and maintenance of keying relationships between authorized parties [41]. In summary, key management integrates techniques and procedures to establish a service supporting [41]:

1. Initialization of system users within network.
2. Generation, distribution and installation of keying material.
3. Control over the use of keying material.
4. Update, revocation and destruction of keying material.
5. Storage, backup/recovery and archival of keying material.
6. Bootstrapping and maintenance of trust in keying material.

Authentication is the basis of any secure communication. Without a robust authentication mechanism in place the remaining security goals (see Section-2.3.1) are in most instances not achievable. Authentication can only be realized by means of verifying something known to be associated with an identity. In the electronic domain the owner of the identity must have a publicly verifiable secret associated with its identity, otherwise, the node can be impersonated.

Authentication in general depends on the context of usage [41]. Key management is concerned with the authenticity of the identities associated with the six services given above; it is a concept which may seem trivial at first, but one that is not easily achieved. Authentication of users is particularly difficult (and in most network settings impossible) without the help of a trusted authority.

The fundamental function of key management schemes is the establishment of keying material. *Key establishment* can be subdivided into key agreement and key transport [41]. *Key agreement* allows two or more parties to derive shared keying material as a function of information contributed by, or associated with, each of the protocol participants, such that no party can pre-determine the resulting value [41]. In *key transport* protocols, one party creates or otherwise obtains keying material, and securely transfers it to the other party or parties [41]. Both key agreement and key transport can be achieved using either symmetric or asymmetric techniques. A *hybrid* key establishment scheme makes use of both symmetric and asymmetric techniques in an attempt to exploit the advantages of both techniques [41].

3.1.2 Requirements of Key Management Schemes

Key management services should adhere to the following generic security attributes:

Confidentiality: Key management schemes should guarantee key secrecy, i.e. ensure the inability of adversaries or unauthorized parties to learn (even partial) key material.

Key Authentication: Key authentication is a property whereby a communication entity is assured that only the specifically intended and authenticated communication entity may gain access to the cryptographic key material.

Key authentication, in the context of a communication session between two parties, can either be *unilateral* or *mutual*: unilateral authentication means that only one party's keying material is authenticated, while mutual authentication involves validating both parties' keying material.

Possession of the key is in fact independent of key authentication. Key authentication, without knowledge that the intended recipient actually has the relevant key, is referred to as *implicit* key authentication.

Key Confirmation: If key confirmation is provided by a key establishment protocol, communication entities prove possession of authenticated keying material. Key authentication with key confirmation yields *explicit* key authentication.

Key Freshness: The *key freshness* property improves security by ensuring new and independent keys between different communication sessions. By separating communication sessions, the available information for cryptanalytic purposes is limited, which makes cryptanalytic attack more difficult.

Perfect Forward Secrecy: *Perfect forward secrecy* (PFS) ensures that compromise of long-term keys cannot result in compromise of past session keys [46] [61] [41].

Resistant to Known Key Attacks: A key management scheme is vulnerable to *known key attacks* (KKA) if a compromised past session key or subset of past session keys allows [46] [61] [41]: 1) A passive adversary to compromise future session keys and 2) An active adversary to impersonate other protocol participants.

Forward Secrecy: A key management scheme with a forward secrecy property prevents an adversary from discovering subsequent keys from a compromised contiguous subset of old keys [62] [63].

Backward Secrecy: A key management scheme with a backward secrecy property prevents an adversary from discovering preceding keys from a compromised contiguous subset of old keys [62] [63].

Key Independence: Key independence guarantees that a passive adversary who knows a proper subset of keys cannot discover any other keys [62] [63].

Availability: A high availability feature prevents degradation of key management services and ensures that keying material is provided to nodes in the network when expected.

Survivability: Survivability is the capability of the key management service to remain available even in the presence of threats and failures. Survivability goes beyond security and fault tolerance to focus on the delivery of services, even when the system is partly compromised or experiences failures. Rapid recovery of services is required when conditions improve [64]. Survivability includes *byzantine robustness* which implies that the key management service should be able to function properly even if some misbehaving participating nodes attempt to disrupt its operation.

More specifically, key management services with a survivability feature focus on the delivery of essential services (for example certification services in public key infrastructure) and the preservation of keying material (public key certificates, session keys etc.). Survivability can be summarized by the *The Three Rs* [64]:

- *Resistance:* The capability of the system to defend against or tolerate attacks.
- *Recognition:* The capability of the system to detect attacks in process and monitor the extent of the damage or compromise.
- *Recovery:* Recovery is the main feature of survivability. It is the capability to maintain services during attack, limit the extent of the damage and restore full services following the attack.

Robustness: The key management scheme should accommodate hardware and software failures, asymmetric and unidirectional links and network fragmentation/partitioning due to limited/error-prone wireless connectivity [64].

Efficiency: The key management service should be efficient with respect to communication, computational, memory and energy resources.

Scalability: Scalability ensures efficiency as the number of networking nodes rapidly and significantly changes. The key management scheme should thus seamlessly scale to network size.

3.2 Key Management in Mobile Ad Hoc Networks

The main focus of this chapter is to provide a detailed account of the existing peer-to-peer key management schemes for mobile ad hoc networks (MANETs). Investigation by the author within the available publications has led to the classification of the current protocols into the following subsets:

1. Partially distributed certificate authority.
2. Fully distributed certificate authority.

3. Identity-based key management.
4. Certificate chaining based key management.
5. Cluster-based key management.
6. Mobility-based key management.
7. Parallel key management.

Most of the above subsets use public key cryptography due to its superiority in distributing keys, providing authentication and in achieving integrity and non-repudiation [1] [41]. Symmetric key systems need a channel that provides both data integrity and confidentiality: the latter property may not always be readily available without any form of trusted authority or secure side channel (such as an infrared interface).

The *partially distributed certificate authority* group of protocols distribute the trust in the certificate authority to a subset of the network communication entities. The approach mitigates the single point of vulnerability inherent in the *centralized* certificate authority. Protocols considered to represent this implementation method are presented in [1] and [65] respectively (Section-3.3).

The *fully distributed certificate authority* protocol subset preserves the symmetric relationships between the communication entities in MANETs by distributing the burden of key management to *all* communication entities. Each authorized node in the network receives a share of the certificate authority's secret key allowing neighbors to service requests for certification. The protocol that introduced this method is presented in [2] (Section-3.4).

The *identity-based key management* approach borrows concepts from the *partially distributed certificate authority* protocols, but uses an identity-based cryptosystem to reduce the storage requirement compared to conventional public key cryptosystems. The protocol by [66] will be considered as representative of this protocol group (Section-3.5).

In the *certificate chaining based key management* approach communication entities can authenticate certificates by means of finding certificate chains between them. Certificate chaining can be explained by the following example: party A wants to communicate with

party C, which requires party A to authenticate party C's certificate. The two parties have no communication history, but party A trusts the certificate of a third entity, party B. Party B informs party A that it trusts the certificate of party C. Party A which trusts party B will thus also trust party C as a result of party B's recommendation. There is thus a fully connected certificate chain between party A and C through party B which enables party A to authenticate the certificate of party C without any previous communication. Section-3.6 investigates the protocol that introduced the certificate chaining based key management approach as detailed in [67] [3].

The *cluster-based key management* subset relies on a clustering algorithm to subdivide the network into smaller groups. Group members in the same proximity can monitor their neighbors and make recommendations to members from other groups on the authenticity of their neighbors' certificates. The cluster-based subset is introduced by investigating the protocol presented in [4] (Section-3.7).

The *mobility-based key management* subset exploits mobility and node encounters to establish security associations and warrant mutual authentication between nodes. In contrast to the previously discussed subsets, the protocols in this group introduce a shift in paradigm with respect to previous attempts to provide key management for MANETs. Rather than trying to adapt solutions suited for conventional wireline networks, the protocols in this subset use the unique characteristics of MANETs to their advantage. Section-3.8 investigates the symmetric and asymmetric key based protocols that introduced the mobility-based key management approach as presented in [68] [5].

The combination of any of the above key management approaches gives rise to what the author calls the *parallel key management* subset. By integrating two of the above approaches in parallel, the advantages of the one scheme is used to mitigate the disadvantages of the other. This subset can be represented by the scheme introduced in [6], which combines a partially distributed certificate authority [65] and the certificate chaining based key management approach [3] (Section-3.9).

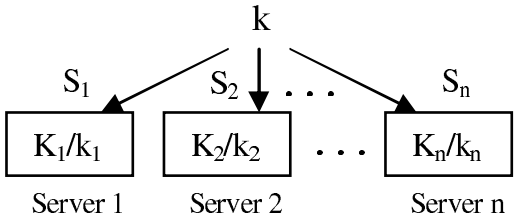

Figure 3.1: Key management service K/k configuration [1]

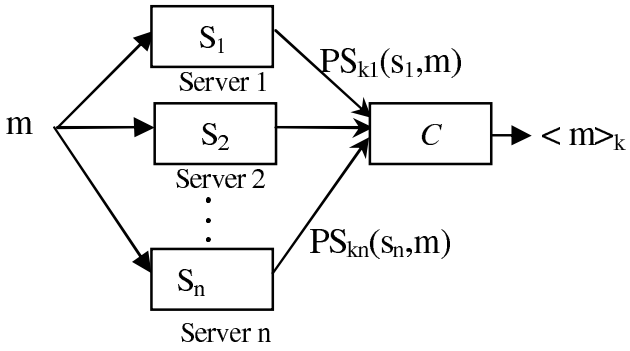

Figure 3.2: Threshold signature K/k generation [1]

3.3 Partially Distributed Certificate Authority Approaches

One of the first approaches to solve the key management problem in MANETs was published in [1]. This approach was later extended in [69] [70] [71] and [65]. Schemes are still actively proposed within this subset [72] [73] [74]. The authors of [1] proposed a distributed public key management service for asynchronous ad hoc networks, where the trust is distributed between a set of nodes by letting the nodes share the system secret. The distributed certificate authority (DCA), illustrated in Figure-3.1 [1], consists of n server nodes which, as a whole, have a public/private key pair K/k. The public key K is known to all nodes in the network, whereas the private key k is divided into n shares $(s_1, s_2, s_3, \ldots, s_n)$, one for each server.

The distributed certificate authority (DCA) signs a certificate by producing a *threshold group signature* as shown in Figure-3.2 [1]. Each server generates a partial signature using its private key share and submits the partial signature to a combiner C. The combiner can be any server and requires at least $t + 1$ shares to successfully reconstruct the digital signature.

3.3.1 System Analysis

3.3.1.1 Initialization Phase

The system as proposed in [1] requires an *off-line* trusted third party (TTP) to construct the distributed public key management service. Prior to network formation the TTP uses a threshold secret sharing scheme [75] to generate shares $(s_1, s_2, s_3, \ldots, s_n)$ of the DCA's private key k. These shares are distributed to n arbitrary nodes (servers) which collectively form the DCA. The TTP must also issue all nodes in the network with the DCA's authentic public key K.

3.3.1.2 Certificate Retrieval

Nodes that require a certificate have to successfully contact at least $t+1$ out of the n DCA servers. As illustrated in Figure-3.2, the threshold signature scheme proposed in [1] makes use of a combiner node C to combine the partial digital signatures from the $t+1$ servers. Any node can be chosen as a combiner, since no extra information about the private key k is disclosed to C. It is always possible for a combiner node to be compromised by an adversary or to be unavailable due to battery depletion or poor connectivity. As a solution the authors of [1] proposed selecting $t+1$ nodes as combiners to ensure that at least one combiner can successfully reconstruct the digital signature. All nodes in the network, including the combiners, can verify the validity of the signature by using the CA public key, K.

The proposal presented in [69] [70] [65] differs from the original proposal in [1], since the threshold signature scheme of Yi and Kravets does not require a combiner node C to construct the group signature. In [69] [70] [65] the DCA is called a MObile Certificate Authority (MOCA). In the MOCA framework the communication pattern is one-to-many and visa versa, which means that a node that requires certificate services needs to contact at least $t+1$ MOCA nodes and receive replies from each of them. The combining of the partial signatures is thus performed by the node requesting certification.

The original proposal in [1] does not specify a communication protocol (certificate retrieval mechanism) for a node to contact the key management service. The proposal in [69] [70] [65] focuses primarily on the one-to-many-to-one communication pattern between a node and the MOCA. The MOCA certification protocol allows a node requiring certification services to broadcasts *certification request* (CREQ) packets. Any MOCA node that receives the CREQ packet answers with a *certification reply* (CREP) containing its partial signature on the certificate. If the node successfully receives $t + 1$ valid CREPs in a fixed period of time, it can reconstruct the full certificate. If the certificate is verified to be correct, the certification request has succeeded. If the number of CREPs is insufficient after expiry of the node's CREQ timer, the process fails and the node can initiate another request. CREQ and CREP are similar to the *route request* (RREQ) and *route reply* (RREP) of on-demand ad hoc routing protocols (for example AODV [33] and DSR [76]). As CREQ packets are routed through nodes, a reverse path is established back to the sender. The reverse path is coupled with timers and maintained for a fixed time period to allow returning CREP packets to travel back to the node requesting the certificate service. In this case an on-demand routing protocol and the MOCA certification protocol can benefit from each other by sharing routing information [65].

Flooding: In the first implementation of the MOCA certification protocol presented in [69], flooding is used for reliable data dissemination. As shown in simulation results presented in [69], the flooding technique yields an effective approach to contact at least $t+1$ servers, but generates a high communication overhead. To prevent the potential broadcast storm nature of flooding, broadcast IDs are used (similar to [33]) such that all the CREQ generated by the same request are tagged with the same ID in order to allow intermediate nodes to drop requests that have already been forwarded.

β-Unicast: To reduce the amount of certification traffic from flooding while keeping an acceptable level of service, the method of β-unicast is introduced in the second report on the MOCA service [70]. The method relies on multiple unicasts instead of flooding, using sufficient cached routing table information. The investigations in [69] show that a node client caches a moderate number of routes to MOCA nodes under reasonable certification traffic in the network. The parameter β represents a sufficient number of cached routes

required by a node to use unicast instead of flooding. Flooding is thus the default method for contacting the MOCA in case the node fails to accumulate enough information in its cached routing tables. If the node perceives the network to be stable with relatively low mobility, $t+1$ cached routes may be sufficient to initiate a multiple unicast CREQs. To guarantee a reply from at least $t+1$ servers the authors in [70] introduce safety margin α. In the case of instability the nodes should send out an additional α CREQ to increase the probability of successfully reaching $t+1$ servers. The sum of α and the crypto threshold t is called the unicast threshold and represented by β. If there are more than sufficient (β) routes in the node's local cache, then the choice of which routes to use will affect performance. Three schemes are suggested in [70]:

- *Random MOCA nodes*: A random number β of MOCA nodes in the routing table are selected.
- *Closest MOCA nodes*: By utilizing the readily available hop count information in the routing table, β MOCA nodes with the minimum hop count are chosen.
- *Freshest MOCA nodes*: The most recently added β routes are used for β-unicast.

3.3.1.3 Certificate Revocation

Certificate revocation is not given much attention in [1] [69] [70] or [65]. The authors propose a simple certification list (CRL) approach. In the MOCA framework $t+1$ nodes must agree to revoke a certificate. Each MOCA node generates a *revocation certificate* that contains information of the certificate to revoke. The MOCA node then broadcasts its partially signed revocation certificate across the network. Nodes that received $t+1$ partial certificates will reconstruct the revocation certificate and update their local CRL.

3.3.1.4 Certificate Renewal

Expired certificates can be renewed by sending a CREQ message to any $t+1$ MOCA nodes. Each MOCA node will update the certificate contents with relevant information

(for example with a new expiry time) and newly bind the public key to the nodes identity by generating a partial signature.

3.3.1.5 Share Update

The key management services in [1] [65] employ proactive key share refreshing [77] to thwart mobile adversaries and adapt to changes in the network. An adversary attempting to break the system must compromise more than the threshold t servers within the time interval between key updates.

3.3.2 Discussion on the Partially Distributed Certificate Authority Approaches

One of the advantages of the distributed certificate authority proposals is that they address the lack of server infrastructure in MANETs by distributing the functionality of a central authority among a group of users. The partly distributed system offers security services with a higher availability feature than a centralized server approach. The scheme however, does not have a certificate revocation or synchronization mechanism to update servers.

The solution presented in [1] has remnants of its wired predecessors, namely a trusted authority, specialized server and combiner nodes. The MOCA framework proposed in [65], in contrast to the original DCA proposal [1], does not require a combiner server, C, effectively solving the problem of ensuring the availability of C.

The solution still has a largely centralized approach, although the threshold scheme allows t DCA servers to be compromised without sacrificing the key management service. One of the major assumptions of the solutions proposed in [1] [65] is the presence of an *off-line* TTP that initially empowers servers or distributes keying material before network formation. The information distributed by the off-line TTP makes the solution non-scalable since all network certificates must be known *a priori* by the DCA servers in order to provide access control to certification services. The assumption of an off-line TTP also makes the scheme unsuitable for fully self-organized MANETs.

The communication overhead introduced by [1] [65] is a point of concern as nodes need to contact the DCA every time they require certification. Any node can contact the DCA by flooding the network with a certification request. The flooding method is an effective way to contact t out of n DCA nodes [65]. The problem is that each of the DCA nodes responds with a certification reply causing a reverse packet storming effect of $\mathcal{O}(n)$ [65]. Yi *et al* [65] make a noteworthy effort to optimize the scheme by investigating different data dissemination techniques for contacting the DCA by means of simulation. In another paper, [78], the authors of [65], investigate *manycast* which appears to be a promising technique to contact only a subset of members from a group with minimum communication overhead. *Manycast* yields better performance than the flooding and β-unicast techniques discussed in Section-3.3.1 especially if *manycast* is integrated into the routing protocol.

Analysis by the author on these schemes have shown that they are subject to the following weaknesses:

- The distribution of the system's private key is normally performed with a threshold secret sharing scheme [75]. Central to the security of the threshold cryptosystem is the choice of the security parameters (n, t), where n is the total number of nodes forming the DCA and t, the threshold of nodes that must be compromised to render the system insecure. The process of choosing appropriate parameters (n, t) is a very difficult task. The choice of these parameters inevitably forces a trade-off between the security of the system and the availability of the DCA nodes.

  Some of the factors to consider when choosing (n, t) are: the networking environment of the MANET; physical security of the users' nodes; bandwidth requirement/utilization of the users; frequency of certification requests to the DCA; mobility patterns of users; capabilities of attackers and; the availability of the DCA nodes. For example, MANETs are subject to error-prone wireless connectivity, limited energy resources (poor battery life) and limited transmission ranges (see Section-2.1). Nodes forming the DCA may thus frequently be unavailable for the rendering of certification services. A hostile environment, such as those found in military applications, would require t to be set as high as possible. The problem however is that

increasing t up to a "safe" point may prevent users from successfully contacting t out of n DCA nodes. The fundamental observation is that taking all these factors into consideration when choosing (n, t), while keeping in mind the security/availability trade-off, is a difficult problem to solve and is central to the security provided by threshold cryptosystems. In the view of the author, a strong security argument is not possible with this approach.

- In threshold cryptosystems it is impractical to assume that a *mobile* adversary cannot compromise more than t shareholders during the entire lifetime of the system [1] [77] [2]. This forces the nodes forming the DCA to execute a *share renewal protocol* [77] within a variable period T; the choice of T is influenced by similar factors to those that affect the selection of (n, t). The share renewal protocol has to be fully distributed as the DCA members have no access to a central on-line TTP.

- The initial access structure $\Gamma_P^{(n,t)}$ of the share distribution scheme will not remain constant [79]. Assuming the same shareholders to be present at all times are unrealistic and the security/availability trade-off will also have to be altered (by reselecting (n, t) on the fly) as a function of system vulnerability, changing networking environment and current functionality of the cryptosystem. Users may randomly join or leave the DCA group, hence the DCA will exhibit *dynamic* group membership as associated with dynamic peer groups [46]. The security parameter (n, t) will have to be adjusted to allow for the dynamic membership. This obligates the DCA nodes to execute a *secret redistribution protocol* [79] [80] in order to distribute the shares to a new access structure $\Gamma_{P'}^{(n',t')}$ on each membership change or security/availability trade-off adjustment.

- Analysis by the author has shown that the overhead associated with fully distributed secret update and secret redistribution protocols may not be practical in MANETs. The author's studies on existing, fully distributed secret update and secret redistribution protocols [80] [77] [79], have shown that distributed share update/rekeying schemes have a high communication and computational cost, which will have to be executed more frequently in MANETs than expected. To put the overhead in more quantitative terms, share updating requires $O(n+t)$ messages from each DCA mem-

ber, while each DCA member performs $O(nt)$ exponentiations and generates $O(t)$ random numbers. The share renewal protocol has similar overhead with $O(n' + t')$ messages for each DCA member and $O(n't')$ exponentiations and $O(t')$ random numbers generated by each member of the new access structure $\Gamma_{P'}^{(n',k')}$. The network as a whole has a communication cost of $O(n^2 + nt)$ and $O(n'^2 + n't')$ for secret update and redistribution respectively. In the analysis of the schemes a synchronous broadcast system was assumed. It is generally agreed that network wide synchronization is not easily achievable in MANETs. Without a synchronous broadcast system it is not clear if fully distributed secret update and redistribution protocols are possible while defending against all the known attacks [81] [82].

To the best of the author's knowledge there is currently no secret sharing, secret update and secret redistribution schemes available that are suitable for MANETs.

- Nodes in MANETs have a symmetric relationship, hence nodes are all equal and therefore should fairly share or equally distribute the responsibility of providing all network functions. This is not only important for security reasons, but to enable the network to ensure reliable and available services that places the same burden on the computational, memory and energy resources of all the network participants. The use of a DCA as on-line TTP violates the symmetric relationship between network nodes. When the DCA is formed by all the nodes in the network, as in [2], the symmetric relationship is preserved. This however impairs the overall security of the system since an attacker can compromise *any* t nodes in the entire network to break the threshold cryptosystem (see Section-3.4).

  If there exists *heterogeneity* among network participants, the use of nodes with more advanced resources for the DCA nodes is not only unfair, but promotes selfishness or denial of service attacks. Unequally shared responsibility is also fundamentally against the notion of fully distributed systems and motivates localized areas of vulnerability. It is however noted that the notion of heterogeneity can be exploited in network settings where the nodes do not have symmetric relationships, such as those found in military type networks [65]. In such a scenario the burdened nodes will tolerate the exploitation for the benefit - or "survival" of the network as a whole.

- Another issue that has also been surprisingly overlooked is how the members of the DCA will collaborate to sign certificates. This is a more difficult problem than one may think: designing group signatures is much more complex than single party signatures. What is needed by the *distributed authority* based schemes proposed in [1] [65] [2] is a *threshold-multisignature* scheme [83] [84] [85] [86]. *Threshold-multisignature* schemes can be differentiated from threshold *group* signature schemes [87] by the fact that, by definition in the latter, the individual signers remain anonymous. In threshold *group* signature schemes it is computationally difficult to derive the identities from the group signature with the exception of the group manager(s). In contrast, in *threshold-multisignature* schemes, the individual signers are *publicly traceable* and do not enjoy anonymity. Consequently, the traceability property of threshold-multisignature schemes allows the individual signers to be held *accountable* in the public domain. The author believes traceability of the individual signers is essential in MANETs. The current state of the art threshold multisignature schemes [83] [84] [85] [86] are notoriously flawed [88] [89] [90] [91]. What is even more discouraging is that all the existing group signature schemes that have been proposed to date are for conventional networks. It is widely known that solutions suitable for MANETs, in most cases, require a shift in paradigm in order to mitigate the consequences of the unique characteristics of MANETs.

- The existing solution in the *partially distributed certificate authority* subset does not aim to break the routing-security interdependency cycle [92].

3.4 Fully Distributed Certificate Authority Approaches

In [93] [2], a public key management solution is proposed based on the approach originally presented in [1]. Their solution also uses an (n, k)[1] threshold signature scheme to form a distributed certificate authority (DCA). They enhance the availability feature of [1] by choosing n to be *all* the nodes in the network. The private key SK of the DCA is thus shared among *all* the nodes in the network and enables a node requiring the service of the

[1] In this section k will uses, instead of t, for the threshold paramenter notation to be compatible with [2].

DCA to contact any k one-hop neighbor nodes. In contrast to [1], no differentiation is made between server and client nodes with respect to certification services. The solution includes a share update mechanism to prevent more powerful adversaries from compromising the certification service. One of the latest proposals within the fully distributed subset can be found in [94].

3.4.1 System and Adversary Model

The network model considers an ad hoc wireless network with insecure, error-prone and bandwidth constrained communication channels. The network has no infrastructure and a dynamically changing topology. The following assumptions are made:

1. Each node v_i has a unique identifier (ID) and is able to discover its one-hop neighbors.
2. Each node holds a valid certificate signed by the DCA private key SK binding its ID to a public key P_{v_i}. A certificate signed by SK can be verified by the authentic public key of the DCA PK.
3. One-hop communication is more reliable than multi-hop.
4. Each node has more than k one-hop neighbors at any time instance.
5. Detection of node misbehavior is easier and more practical among one-hop neighbors in contrast to multi-hop nodes.
6. Mobility of the network is characterized by the speed of the node with the highest speed S_{max}.

The proposal in [2] [93] attempts to alleviate two types of attacks: denial of service (DoS) and adversary intrusion or node break-ins. Adversaries may issue DoS attacks on various layers of the network stack.

As defined in [77], adversaries can be characterized by one of two models:

Model I: During the entire network lifetime an adversary cannot successfully attack and compromise more than k nodes.

Model II: If the network's lifetime is divided into time slots T, an adversary cannot successfully attack and compromise more than k nodes within T.

The authors of [2] [93] attempt to defend against *Model II* adversaries with a scalable parallel share update mechanism.

3.4.2 System Analysis

3.4.2.1 Initialization Phase of Localized Certification Service

The system as proposed in [2] [93] requires an *off-line* trusted third party (TTP). The RSA based design has a system DCA with an RSA key pair $\{SK, PK\}$. *Prior* to network formation the TTP distributes a certificate signed with the DCA private key SK to each node. The certificate is a binding between the nodes' unique *ID* and public key which may be verified with the DCA's authentic public key PK known by all nodes in the network. It should be clear that the localized certification service never creates or issues an initial certificate. Its functionality is the renewal of expired certificates or re-issuing of certificates thought to be compromised by adversaries.

The off-line TTP distributes the first k shares. In more accurate terms, the TTP distributes to the first k nodes in the network a polynomial share P_v of the certificate signing exponent SK according to a random polynomial $f(x)$ such that $P_v = f(v)$. Upon completion of this task the TTP has no further part in network operation.

The polynomial $f(x)$ can be defined as: $f(x) = SK + \sum_{j=1}^{k-1} f_j x^j$ where f_1, $f_2, \ldots$, f_{k-1} are uniformly distributed over a finite field F.

3.4.2.2 Localized Self-initialization

In [93], the authors propose an algorithm used to distribute shares of SK to nodes joining the network. A node joining the network must have a certificate binding its ID and public key signed by SK. Only a joining node with a valid certificate, hence verifiable with PK, can obtain a share of SK. Since their architecture is built on Shamir's threshold secret sharing scheme [75], only a coalition of k nodes can issue the uninitialized node with a share of SK.

The self initialization protocol can be summarized in four steps:

1. A joining node broadcasts to neighboring nodes a service request with additional local coalition information.
2. Each neighbor (coalition member) selects a *nonce* (random number) for each other node if its ID is *lower* than the other node's ID. A complete *shuffling scheme* is used by the nodes to exchange these nonces between them. In the peer-to-peer exchange of nonces, the nonces are negated by the node with the *lower* ID. The nonces are encrypted with the public key of the intended recipient coalition member.
3. The encrypted shuffling packets or nonces are then routed to coalition members.
4. Each of the coalition members computes a shuffled partial secret share SS from its partial share P_v in SK. Since it is possible to derive SS from P_v, SS is blinded to node v_x by adding the sum of all nonces to SS. After decrypting the nonces each coalition member computes $SS' = SS + \sum nonces$ and transmits its computed partial share SS' to v_x.

3.4.2.3 Certificate Issuing and Renewal

As described in the *initialization phase* each node holds a share of the private key SK according to the random polynomial $f(x)$. In [93] node v_i firstly locates a coalition β of k neighbors $\{v_1, \ldots, v_k\}$ and broadcasts certification requests to the selected neighbors.

Each node $v_j \in \beta$ consults its monitoring data and makes a decision to grant or refuse certification. If v_i is certified as legitimate, v_j returns a partial certificate P_{vj} to the requesting node v_i. The node then recovers the certificate as a whole from the partial certificates using the *k-bounded coalition* offsetting algorithm as given in [93]. This scheme remains functional even if coalition members are under attack since only k partial signatures are required from neighboring nodes.

In the later paper [2] two drawbacks are identified with the approach in [93]:

1. If any of the nodes $v_j \in \beta$ fail during the requesting process, all the other partial certificates become useless.
2. When v_j receives a request from v_i the monitoring records may not provide sufficient information to grant certification. This will be the case in a network with high mobility where v_j and v_i have never met before or have had insufficient interaction.

In order to solve the first drawback, *dynamic coalescing* is introduced. This method stems from the observation that the coalition can be dynamically formulated from *any* responding nodes, instead of being specified *a priori* by the requesting node v_i. Figure-3.3 [2] illustrates how dynamic coalescing might overcome the first problem identified above.

To solve the second drawback and accommodate mobility, certification is granted if no bad records are found. The unavailability of records is thus taken as insufficient reason to deny certification.

3.4.2.4 Certificate Revocation

Certificate records maintained by node v_j consist of two components: monitored data (certificates and behavior) of neighboring nodes and a certificate revocation list (CRL). The CRL is a list containing user IDs and accusers. If a node v_j concludes by direct data monitoring that a neighboring node is compromised, it marks the node "convicted" in its own CRL. The accuser v_j also floods the network with a signed accusation against

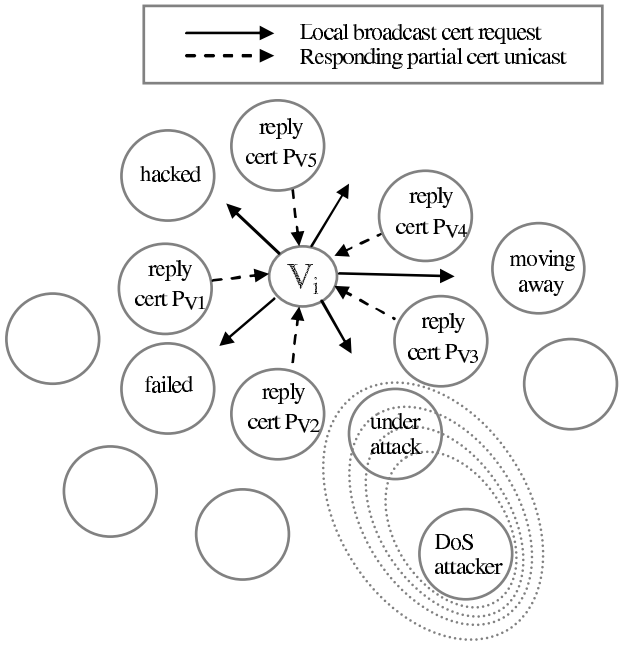

Figure 3.3: Dynamic Coalescing [2]

the node. Now assume node v_j receives a signed accusation against an accused node, it checks in the CRL if the node has been previously marked "convicted". If this is the case the message is regarded as being a confirmation of the conviction and dropped; if not, the node is marked as "suspect". To avoid the conviction of legitimate nodes at least k accusations against a node is required before it is marked "convicted".

3.4.2.5 Parallel Share Updates

For a share distribution system to be robust against *Model II* adversaries, periodic updates are required [77]. This prevents an attacker from compromising more than k secret shares in the period between periodic share updates. The proposal in [93] gives two approaches to achieve share updates. The first approach is a process based on the localized self-initialization explained in Section-3.4.2.2. The second approach features parallel share updates with faster convergence. The update is performed by distribution of a new random polynomial $f_{UPDATE}(x)$ whose coefficients are encrypted with SK to ensure authenticity. The node's new share in SK can then be collaboratively evaluated as $P_{vUPDATE} = f_{UPDATE}(v_i)$ by k neighboring nodes. Each neighbor returns their partial

share update to v_i in a manner similar to that of the certification service. The parallel share update approach thus consists of three steps: collaborative generation of the update polynomial $f_{UPDATE}(x)$; robust propagation of the update polynomial's coefficients and distributed evaluation of share updates.

3.4.3 Discussion and Comments on Fully Distributed Certification Authority Approaches

The fully distributed certificate authority proposal represents an improvement on the original partially distributed CA scheme in [1] by fairly distributing the burden of holding a share of the secret key between all nodes in the network. This effectively preserves the symmetric relationship between network participants. The availability of the key management service is increased as now any k nodes, in a local neighborhood, can renew or issue a certificate. Nodes not in possession of a share can also contact at least k nodes to obtain a share.

Although innovative in design, the proposal suffers from most of the weaknesses discussed in Section-3.3.2. As in the original proposal [1], a trustworthy *off-line* authority must issue every node before network formation with a certificate, binding a unique ID to a public key. This requirement makes the proposal impractical for fully self-organized ad hoc networks. Similar to [1] [65] this solution is also non-scalable since all identities must be known *a priori*.

Since the number k is a trade-off between security and availability, the solution requires a way of adjusting k as the network expands. As pointed out in [3] it is not clear how k will be adjusted in a network with a rapidly increasing or decreasing node density. The increase in availability of the certification service also comes at the cost of security since *any* k nodes in the network can be compromised to break the system; the author does not believe that the assumption of a Model II adversary (Section-3.4.1) is realistic if n spans the entire network. For this reason k must always be chosen large enough to ensure the security of the system

The assumption that each node will always have at least k one-hop neighbors is limiting: nodes may find themselves with less than k neighbors more frequently than expected.

As in the case of [1] [65], it is not clear if this approach can break the routing-security interdependency cycle [92].

3.5 Identity-Based Key Management Approaches

ID-based cryptography [95] [96] [97] originated from the need to reduce the memory storage cost of public key encryption keying material and the burden of obtaining explicit public keys. Public keys in an ID-based scheme are nothing other than the identities of the users themselves. The identities, which are publicly known data, must uniquely identify the users. ID-based schemes thus uniquely and privately bind secret keys to identities.

The identity-based signature schemes are normally specified by four randomized algorithms [96]:

1. *Setup*: The setup algorithm takes as input a security parameter and returns a master public/private key pair K_M/k_M for the system. The master private key is only known by the trusted third party (TTP) or private key generator (PKG) of the system.

2. *Extract*: The extract algorithm takes as input the master secret key and an identity ID and returns the personal secret key corresponding to the ID.

3. *Encrypt*: The encrypt algorithm takes as input the master public key K_M and the identity ID of the recipient and a message and returns the corresponding ciphertext. Note that ID serves as the public key of the recipient.

4. *Decrypt*: The dencrypt algorithm takes as input the master public key, a ciphertext and the personal secret key and returns the original message encrypted with the ID corresponding to the personal secret key.

The personal secret keys in an identity-based cryptosystem can also be seen as an *implicit*

symmetric key certificate, i.e. the personal secret key is encrypted with the master secret key of the PKG.

3.5.1 System Model

In [66], identity-based cryptography [95] [96] is combined with threshold cryptography [41] to avoid the extensive computational cost of public key cryptography. This solution is similar to [1] with the threshold certificate authority replaced by a *threshold* private key generator (PKG). For this reason details of the proposal are not given and readers are referred to [1] (Section-3.3). In [66], a master public key (PK^*) is generated by all users on network formation. When users want to encrypt messages, they use their identities as their public keys and obtain their corresponding private keys by contacting at least t out of n nodes with each share of their private keys encrypted by a share of the master private key (SK^*). After decrypting each share with the master public key, combining all t shares yields the private key corresponding to their identity.

Another example of a protocol within the identity-based key management subset can be found in [98].

3.5.2 System Analysis

3.5.2.1 Initialization Phase

On network formation, users agree on a key issuing policy and exchange all relevant security parameters. These must be mutually acceptable and nodes not approving of these parameters may choose to abort the network formation process. The initial set of nodes then form a *threshold private key generation service* (PKG), which generates a master public/secret key in a distributed manner. The master private key is thus distributed to n nodes, each holding a share of SK^*. An adversary with less than threshold t shares cannot recover the master private key. The master public key in turn is given to all joining members of the network.

3.5.2.2 Registration Phase

After the initialization phase the PKG can start issuing users with their private keys based on their identity and the key issuing policy. The node contacts at least t of the nodes forming the PKG, which each reply with their part of the requesting node's private key. Upon receipt of t correct shares, the user can compute its private key.

3.5.3 Discussion and Comments on Identity-Based Key Management Approaches

In the initial phase of the proposed scheme [66], nodes decide on a mutual set of security parameters. Any node that is not satisfied with the choice of parameters can choose not to participate in the network. The authors of [66] state that their scheme is independent of the initial negotiations. This independence is difficult to see as it is the responsibility of the key management protocol to successfully initialize all system users within a domain [41] (see Section-3.1.1). If adversaries are able to influence the selection of system parameters they will be able to force nodes not to participate in the network.

The proposal does not address the issue of how the initial set of nodes will form the PKG or how a node will obtain a private key from the *distributed* PKG. Distribution of the master private key, as mentioned in [66], can be done using the algorithm presented in [99]. Integration of the distributed key generation scheme [99] into the *setup algorithm* of the identity-based signature scheme [97] will enable the generation of a distributed master private key. A problem is noted with the implementation of *extract algorithms* of existing identity-based signature schemes in distributed systems. It is noted that the *extract algorithms* in [97] [96] [95] are designed for an entity obtaining a personal private key from a *centralized* PKG. Any centralized service in ad hoc networks is a single point of vulnerability. The *extracting algorithms* will have to be modified for negotiation of a private key with a *distributed* PKG. See observations in Section-3.3.2 on secret sharing, secret update, secret redistribution and threshold multisignature schemes.

The proposal does not avoid the weaknesses of ID-based cryptography. The major problem

with ID-based cryptographic schemes is that they yield only *level* 1 trust [100], i.e. the private key of users is known by the trusted authority. In conventional networks this is not so much of a problem, but in MANETs where the trusted authority is distributed between *on-line* servers or emulated by an arbitrary *off-line* entity, this may not be feasible.

Assume the node can negotiate a personal private key with a *distributed* PKG; a major problem is how the PKG will securely transfer to the requesting node its personal private key shares. In the scheme proposed in [66], the requesting node shares no secret with the PKG, for example a common symmetric key, nor do the nodes have public/private key pairs. It is therefore not clear how a node will obtain its personal key from the PKG in the presence of an adversary. This problem can only be solved by setting up some secure channel or by pre-distributing common keying material, neither of which is ideal in ad hoc networks.

The proposal swaps one difficult problem for another. In the case of a PKI solution, the primary concern is the authentication of public keys. In [66] the problem is to authenticate the identity of a node before sending the shares of the personal private key corresponding to the identity.

The solution is vulnerable to a *man-in-the-middle* attack on nodes joining the network [66].

Finally this approach cannot break the routing-security interdependency cycle [92].

3.6 Certificate Chaining Based Approaches

One of the most recent proposals presented in [3] takes a step closer to meeting the constraints of MANETs. Unlike previous solutions, the public key infrastructure (PKI) in this proposal does not require *any* trusted third party. This makes the scheme suitable for fully self-organized MANETs. Each node issues its own certificates to other nodes in a manner similar to Pretty Good Privacy (PGP) [101]. It differs from PGP in the fact that there are no centrally managed certificate directories (on-line certificate servers), but certificates are rather stored and distributed by nodes in a self-organized nature.

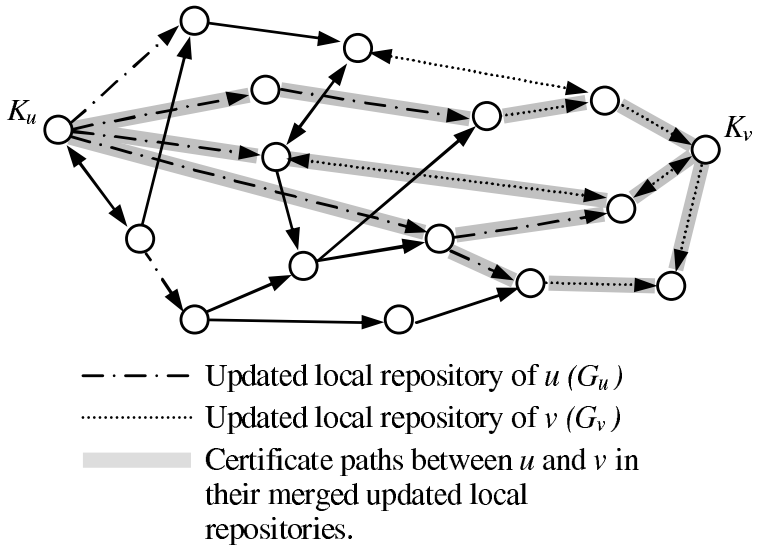

Figure 3.4: A certificate graph and certificate paths between users u and v in their merged updated local repositories [3]

Each node keeps a limited certificate repository comprising of certificated nodes in its local neighborhood. As illustrated in Figure-3.4 [3], when a node u wants to validate the certificate of another node v, the nodes combine their certificate repositories and u attempts to find a chain of valid public key certificates between them.

3.6.1 System Model

The authors of [3] present their scheme in terms of an abstract model. In their model, the public keys and the certificates of the system are represented as a directed certificate graph $G(V, E)$, where V and E stand for the set of vertices and the set of edges, respectively (Figure-3.4). The vertices of the certificate graph represent public keys and the edges represent certificates. More precisely, there is a directed edge from vertex K_u to vertex K_w if there is a certificate signed with the private key of u that binds K_w to an identity. A certificate chain from a public key K_u to another public key K_v is represented by a directed path from vertex K_u to vertex K_v in G. Thus, the existence of a certificate chain from K_u to K_v means that vertex K_v is reachable from vertex K_u in G (denoted by $K_u \longrightarrow_G K_v$).

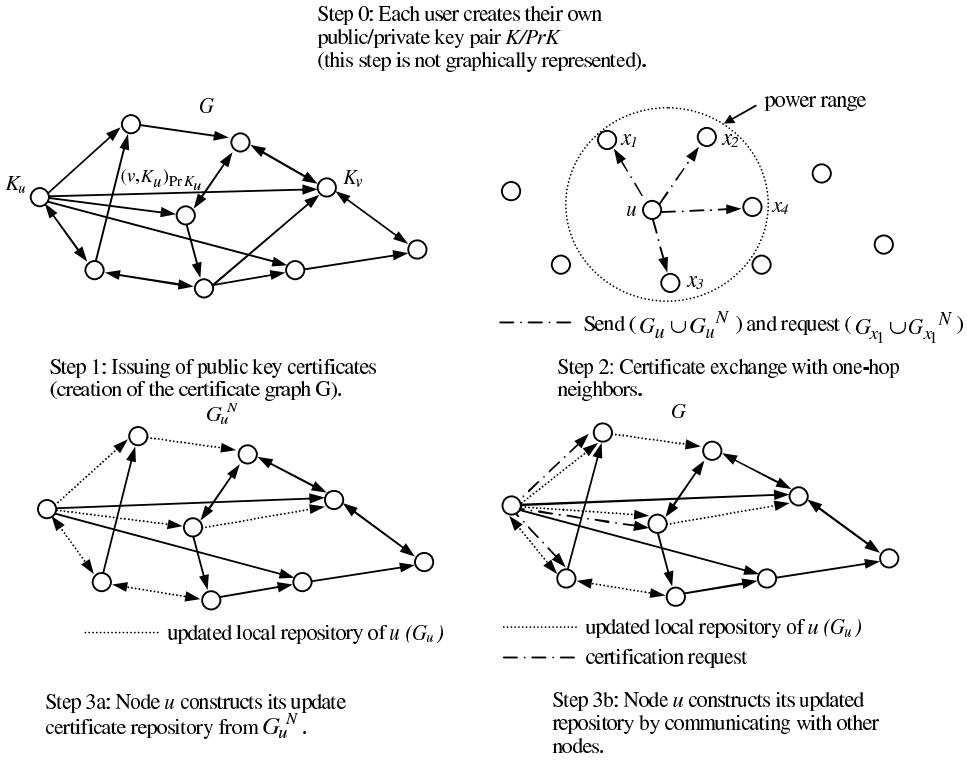

Figure 3.5: Four steps in initial phase of certificate chaining proposal [3]

3.6.2 System Analysis

3.6.2.1 Initialization Phase

The initial phase of the system is executed in four steps: each node creates a public/private key pair; each node creates a self-certificate, issues certificates to other nodes and constructs an nonupdated certificate graph; nodes exchange certificates; and create updated certificate repositories. Each of these steps is illustrated in Figure-3.5 [3] and explained in more detail in the sections that follow. Note that the step numbering is kept consistent with the numbering used in [3].

3.6.2.2 Step-0: Creation of Public/Private Key Pairs

Similar to PGP [101] users locally create their own private key and corresponding public key.

3.6.2.3 Step-1: Creation of Self-certificates and Issuing of Public Key Certificates

Individual public key certificates are issued by the users themselves with a limited validity period. When a node u has confidence in the public key/identity binding of node v, node u can issue a certificate (recommendation) to vouch for the binding. The nodes use these certificates to start constructing a certificate graph G.

3.6.2.4 Step-2: Certificate Exchange

The certificate exchange mechanism allows users to share and distribute certificates in their repositories. Certificates are stored at least twice: by the issuer of the certificate and by the user to whom the certificate is issued. The certificate exchange process consists of the periodic exchange of certificates between nodes and their neighbors. Exchanges are asynchronous. Users send their updated subgraphs G_u and nonupdated subgraphs G_u^N to their neighbors, who use the certificates to create or expand their nonupdated subgraphs. The message contains only the hash-value of the certificates. The node checks the hash-values against those held in its repositories and requests only the certificates with a negative hash-value comparison. The certificate exchange process has a low communication cost since certificate exchanges are only performed locally in a one hop fashion.

3.6.2.5 Step-3: Construction of Updated Certificate Repositories

The nonupdated repositories (subgraphs) provide the nodes with only an incomplete view of the certificate graph. An updated certificate repository is constructed by node u by selecting a subgraph G_u of G. This can be performed in two ways: nodes can use the same local repository construction algorithm to explore only a relevant part of the certificate graph G (Step-3a, Figure-3.5) or construct an updated repository by communicating with their certificate graph neighbors (Step-3b, Figure-3.5).

3.6.2.6 Certificate Revocation

Users can revoke any issued certificate to other users in the instance of distrust in the public key/identity binding. Similarly users can also revoke their own certificate if they believe that their private key has been compromised. The authors of [3] propose two revocation schemes: *explicit* and *implicit*.

In the *explicit* scheme the user issues an explicit revocation message to nodes in its local neighborhood. This will inevitably also be the users that usually request certificates from the revocation node under normal operation.

The *implicit* revocation scheme is based on the expiration time of the certificates. The scheme assumes that users will establish communication within the validity period of the certificate and exchange an updated version of the certificate.

3.6.3 Discussion on Certificate Chaining Based Approaches

The *fully* self-organized key management scheme, presented in [67] [3], was the first to give an indication that key management, without any form of TTP, may be possible in MANETs. The self-organized scheme is clearly an advance on previous efforts [1] [65] [2] [66] [4] to provide key management for MANETs: eliminating any form of on-line TTP eliminates most of the problems associated with these schemes (see Section-3.3.2 for a discussion on the distributed certificate authority proposals). The *fully* self-organized scheme presented in [67] [3] was designed for "open" MANETs and therefore not truly comparable with the schemes that assume an off-line trusted authority.

In [67] [3] the authors address a very difficult problem. How does one explicitly authenticate the public key of a user without any form of off-line or on-line trusted authority? In MANETs this problem has to be solved with sporadic connectivity, while optimizing communication and computational resources. The problem can alternatively be defined in terms of trust establishment. If a user A trusts the certificate of another user B, user A has confidence that the public key contained in the certificate belongs to user B. User

A and B therefore have a *direct* trust relationship. Taking this notion a step further, user A can support the validity of the certificate of user B by signing the certificate with its own private key; any other user in the network that trusts the certificate of user A will also trust the certificate of user B if they are able to verify the recommendation of user A, hence verify the signature. If user A also recommends the certificates of other users, a *hierarchical* trust model is created. Certificate chaining used in PGP [101] naturally evolves from a direct and hierarchical trust model combination.

The authors of [67] [3] successfully adapt the certificate chaining authentication approach to MANETs. The main difference between their scheme and PGP [101] is that the latter stores certificates in centralized repositories. In [67] [3] the certificates are disseminated and stored by all nodes without any assistance from a trust authority.

Trust relationships take time to form and require user interaction. The *fully* self-organized scheme [67] [3] therefore introduces a delay in the setup of security associations. As a result the solution may encounter a problem in the initial phase when the number of issued certificates is insufficient to yield a sufficiently dense certificate graph.

Inherently a chain of trust provides *weak* authentication [102] [103]. A common assumption in most distributed authentication protocols is that trust is implicitly transitive [103]. Trust transitivity means, for example, that if Alice trusts Bob who trusts Clark then Alice will also trust Clark. This has been shown to be *generally* untrue [102]. Josang *et al* define a valid transitive trust chain as a chain where every link in the chain contains the same trust purpose [104]. Abdul-Rahman *et al* point out that valid transitive trust chains satisfy four conditions with reference to the example above [103]. It is concluded from [102] [103] [104] that it is very difficult to ensure valid transitive trust chains with more that two links. The author feels that users (in general) are not able to make intuitive security related decisions. The public's lack of even the most basic knowledge (such as what is a public key certificate) makes any scheme that relies on users *reasoning* about security, vulnerable to attack. Examples of schemes that avoid such "user reasoning" are presented in [5] [105] [106]. In a two link trust chain Alice has a direct trust relationship with Bob and relies on recommendations from Bob based on this direct trust relationship. In

practice a direct trust relationship implies that Alice and Bob know each other personally. This means that Alice can check with Bob if they share the same trust purpose or trust conditions. If the chain is extended to three links (four users) then Alice will have an indirect trust relationship with Clark, hence may not know him personally. The indirect trust relationship lessens Alice's ability to ensure that she and Clark have the same trust conditions.

Futhermore, a chain is as strong as its weakest link. If any node along the chain is compromised or subject to byzantine behavior it may result in false authentication. In "open" or *fully* self-organized networks this may be even more relevant than in "closed" networks since an adversary does not have to compromise nodes to participate in the certificate exchange mechanisms.

3.7 Cluster-Based Key Management Approaches

The key management scheme proposed in [4] originates from the certificate chaining approach [3]. The authors assume a cluster-based network model constructed with the *zonal algorithm* [107]. The zonal algorithm for clustering ad hoc networks, partitions the network into different subsets using a distributed algorithm for finding the minimum spanning tree (MST). Once the network is partitioned and the MST determined for each subset, the algorithm computes the weakly connected dominating sets of the regions. Finally, it fixes the borders of the clusters, i.e. connects unjoined regions, by the inclusion of additional nodes in the sets. Nodes clustered together in the same region form a group and are assigned a unique ID. The nodes learn the group IDs of other nodes by exchanging messages.

3.7.1 Trust Model

Each user is responsible for the creation of their own public/private key pair and generation of a self-certificate. Any node can sign the public key certificate of another node in the *same* group upon request. Nodes are assumed to have some monitoring components

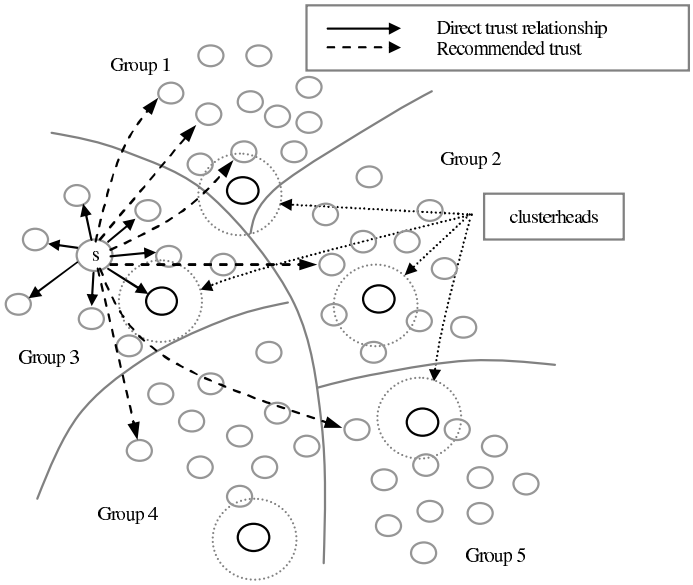

Figure 3.6: Cluster-based Trust Model [4]

that enable them to observe the nodes' behavior in their group and assign each node a trust value. The trust value is defined as an authentication metric, which represents the assurance with which a requesting node s can obtain the correct public key of a target node t. The trust between nodes in the same group is referred to as *direct* trust. The trust relationship between nodes in different groups are referred to as *recommendation* trust. The trust model is shown in Figure-3.6 [4]. Each node should thus have a *trust table* for storing the trust values and related public keys of nodes it knows in the network.

3.7.2 Public Key Certification and Trust Value Update

If a node s wants to obtain the public key certificate of some other node t, s performs the following procedure [4]:

1. Node s looks up the group ID of t, denoted as φ_t.
2. Node s consults its trust table and sorts the trust values of the nodes known to s in group φ_t. Let $i_1, \ldots, i_n \in I$ where i_n denotes the node with the highest trust value.
3. Node s then sends certification request messages to each node in subset I. These

nodes are also referred to as the *introducers*.

4. Node s collects the reply messages $m \in M$ from I where $m = \{P_{k_t}, V_{i_k,t}, \ldots\}Sk_{i_k}$. P_{k_t} denotes the public key of t, $V_{i_k,t}$ denotes the trust value from i_k to t. Sk_{i_k} denotes the secret key of i_k which is used to generate a signature on m.

5. Node s compares the public keys received from I and follows the majority votes. Let $i_{good} \in I_{good}$ and $i_{bad} \in I_{bad}$, where I_{good} are the nodes thought to be honest and I_{bad} the remaining perceived dishonest nodes. Node s perceives the public key of t received from I_{good} to be authentic.

6. The trust values of the nodes $\in I_{bad}$ are reduced to zero. Node s then computes and updates the trust value of t using the following equation, where i_k denotes the nodes and n the number of nodes in I_{good}:

$$V_{s,i_k,t} = V_{s,i_k} \bigodot V_{i_k,t} = 1 - (1 - V_{i_k,t})^{V_{s,i_k}} \tag{3.1}$$

and

$$V_t = 1 - \prod_{k=1}^{n}(1 - V_{s,i_k,t}) \tag{3.2}$$

The public key certification and trust value update procedures are illustrated in Figure-3.7 [4] and Figure-3.8 [4] respectively.

Step 6 may need some further explanation. After receiving, decrypting and comparing the trust values of I in step 5, s can calculate the new recommendation trust relationships from s to t via the nodes in $i_k \in I$ using Equation-3.1. Note that the nodes $i_{bad} \in I_{bad}$ make no contribution since their trust values are reduced to zero. The $\bigodot$ operator is defined in [108] and is given as $V_1 \bigodot V_2 = 1 - (1 - V_2)^{V_1}$. In [108], derivatives of the formula $V_1 \bigodot V_2$ are used to compute new trust relationships between V_1 and V_2 based on the direct trust values and recommendation trust values between them. See [108] for further details on the origin of $V_1 \bigodot V_2$.

Once $V_{s,i_k,t} \; \forall \; i_k$, has been computed, s can compute the ultimate trust value V_t of t as seen by s after pubic key certification using Equation-3.2.

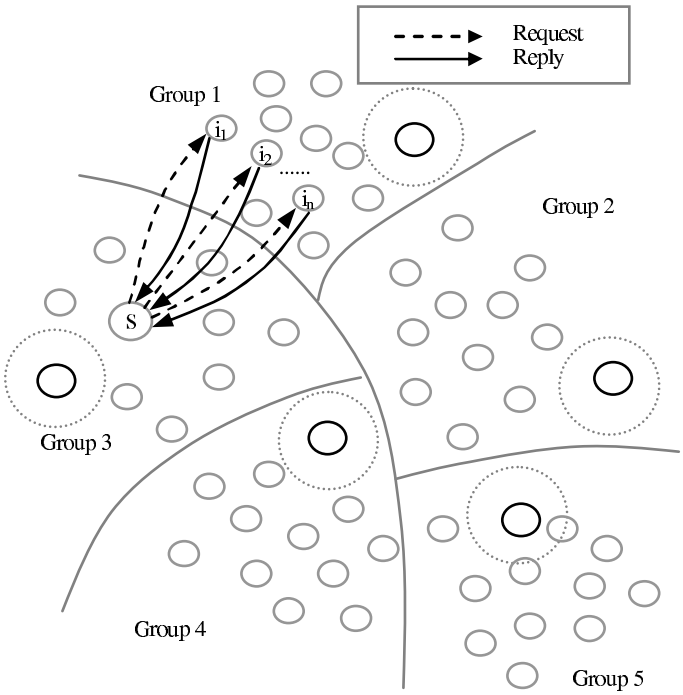

Figure 3.7: Public Key Certification [4]

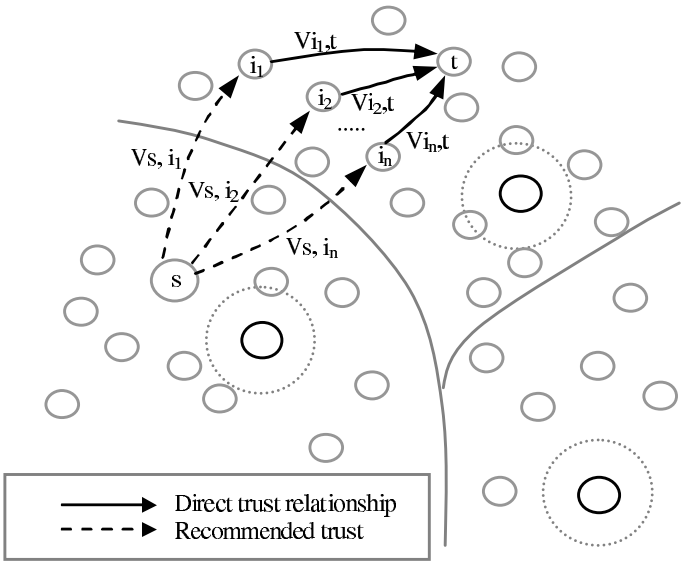

Figure 3.8: Trust Value Update [4]

3.7.3 Discussion on Cluster-Based Key Management Approaches

This discussion will refrain from a detailed evaluation of network clustering in MANETs, but will rather focus on some issues related to the analysis of cluster-based key management schemes.

- When MANETs scale, route calculations become increasingly expensive [107]. The task of routing algorithms may be simplified by confining route discovery to a substructure of the network. By introducing clustering into the network *local* messages can be transmitted on short paths within the same cluster, while *long-distance* messages travel longer distances from cluster to cluster. Each cluster is assigned a *clusterhead* as representative of the cluster. The clusterhead takes on the responsibility of participating in inter-cluster route calculation and long-distance message forwarding. The long-distance message forwarding requires the clusterheads to use more transmission power, which will significantly contribute to the depletion of the node's energy resources [9].

- The dynamic nature of MANETs makes clustering very problematic. The main concern is the selection, configuration, maintenance and replacement of clusterheads. The role of a clusterhead should be able to be performed by any node in the network. Clusterheads are therefore no different from other nodes in MANETs and thus exhibit some similar characteristics. Due to unreliable connectivity and route failures, clusters may also partition. The frequent maintenance of clusterheads and cluster membership is unavoidable and may therefore cause impractical computational and communication overhead.

- The assignment of clusterheads is very difficult since nodes may not voluntarily take on the responsibility [7] or necessarily have the required capacity to accommodate the additional overhead.

- The clusterheads become very convenient central points of attack for adversaries, for example, an adversary which can establish itself as a clusterhead effectively controls the whole cluster. Clearly this will not work in *fully* self-organized MANETs.

The authors of [4] simulated their scheme in GloMoSim [109]. The 100 nodes in the $600m \times 600m$ network were divided into 5 groups with varied mobility between $0 - 10m/s$. The objective of the simulation was to test the public key management scheme in the presence of malicious nodes. A percentage, m, of the nodes were set to be adversaries and therefore always returned false public keys and trust values. The scheme was not implemented in parallel with a clustering algorithm but was divided into *fixed* groups. It was thus assumed that the network clustering structure was already built. If the scheme was to be implemented on top of the clustering algorithm (for example the zonal algorithm [107]) it can be anticipated that the simulation would produce significantly different results. Schemes should always be simulated in a realistic setting, i.e. performance can only be determined if the protocol under investigation is simulated together with the schemes central to its operation. These additional schemes in turn may consume additional network resources which will certainly influence the results obtained from the simulations.

The cluster-based key management scheme does not alleviate the disadvantages of the certificate chaining approach [3] (see Section-3.6.3).

Key management schemes in MANETs should not rely on the functionality and correct operation of other schemes. A cluster-based key management approach relies on the effectiveness of a clustering scheme (for example the scheme presented in [107]). Attacks on the clustering protocol thus introduce additional vulnerabilities and may create an indirect way of compromising the key management scheme and consequently the network's security mechanisms as a whole.

3.8 Mobility-Based Key Management Approaches

In [68] [5], the authors propose mobility-assited key establishment schemes for MANETs. As mentioned before, this author views these schemes as a significant advance in the state of the art: in contrast to the previously discussed subsets, the protocols in [68] [5] introduce a shift in paradigm with respect to previous attempts to provide key management for MANETs. Most of the existing key management schemes for MANETs try to mo-

dify solutions suited for conventional wireline networks which may not always be ideal in MANETs. The proposals investigated [68] [5] are peer-to-peer key establishment schemes that rely on user mobility to bring nodes within each other's transmission range. This allows them to exchange their keying material without relying on a secure routing infrastructure. This effectively breaks the routing-security interdependence cycle [92]. The remainder of the section focuses on the key agreement techniques proposed in [68] [5] for *fully* self-organized MANETs.

3.8.1 System Model

In [68] [5] the authors consider two models: a fully self-organized model and one with an off-line trusted authority. The latter, an authority based approach will not be considered.

If a public key cryptosystem is used, two users u and v share a two-way security association if they exchange their triplets (U, k_u, a_u) and (V, k_v, a_v), where (U, V) are the names of users u and v, (k_u, k_v) are their public keys and (a_u, a_v) their respective node addresses. In a symmetric key setting the public keys are replaced by a shared key k_{uv}

Users are equiped with wireless nodes with an integrated side channel (such as an infrared interface). The side channel is used to setup security associations when users physically meet in the network. This inherently constitutes visual authentication by the users and allows users to bind user names to keying material. The security association setup mechanism can be enhanced through the use of *friend* nodes [5].

To explain the key establishment mechanisms of [68] [5] in more detail, the establishment of security associations in both a public key and symmetric key setting is considered.

3.8.2 System Analysis

3.8.2.1 Public Key Approaches

Figure-3.9 illustrates the three main mechanisms for key establishment proposed in [68] [5]. The first [Mechanism (a)] allows users to establish a security association directly over the secure side channel during a physical encounter. The side channel ensures data integrity by eliminating the active adversary (see Section-2.3.2.3). Coupling Mechanism (a) with key confirmation and a defence against replay attacks results in Protocol 1 [68] [5] detailed below:

Protocol 1 : Mechanism (a)

msg1 (secure side channel)	$u \rightarrow v$:	$a_u \parallel [\xi_u = h(r_u \parallel U \parallel K_u \parallel a_u)]$
msg2 (secure side channel)	$v \rightarrow u$:	$a_v \parallel [\xi_v = h(r_v \parallel V \parallel K_v \parallel a_v)]$
msg3 (radio channel)	$u \rightarrow v$:	$r_u \parallel U \parallel K_u \parallel a_u$
msg4 (radio channel)	$v \rightarrow u$:	$r_v \parallel V \parallel K_v \parallel a_v$
	u:	$h(r_v \parallel V \parallel K_v \parallel a_v) = \xi_v?; V?; match(K_v, a_v)$
	v:	$h(r_u \parallel U \parallel K_u \parallel a_u) = \xi_u?; U?; match(K_u, a_u)$
msg5 (radio channel)	$u \rightarrow v$:	$\sigma(r_v \parallel U \parallel V)$
msg6 (radio channel)	$v \rightarrow u$:	$\sigma(r_u \parallel V \parallel U)$

In msg 1 and msg 2 the users exchange their network addresses (a_u, a_v) and the hash values (ξ_u, ξ_v). Each party computes a hash value by taking the hash $h(\cdot)$ of the concatenation $(\parallel)$ of their random numbers (r_u, r_v) and triples $[(U, K_u, a_u), (V, K_v, a_v)]$. They need each other's address in order to exchange keying material on the radio interface in the following rounds. In msg 3 and 4, users exchange their triplets and random numbers over the radio interface. Each node checks (?) whether the hash of the received random numbers and triplets over the radio link match the received hash values received over the side channel, i.e. $h(r_u \parallel U \parallel K_u \parallel a_u) = \xi_u?$ and $h(r_v \parallel V \parallel K_v \parallel a_v) = \xi_v?$. Each user must also verify that the received user name corresponds to the other party ($U?$, $V?$) and that both nodes can verify if the received node address matches the received public key, i.e. $match(K_u, a_u)$

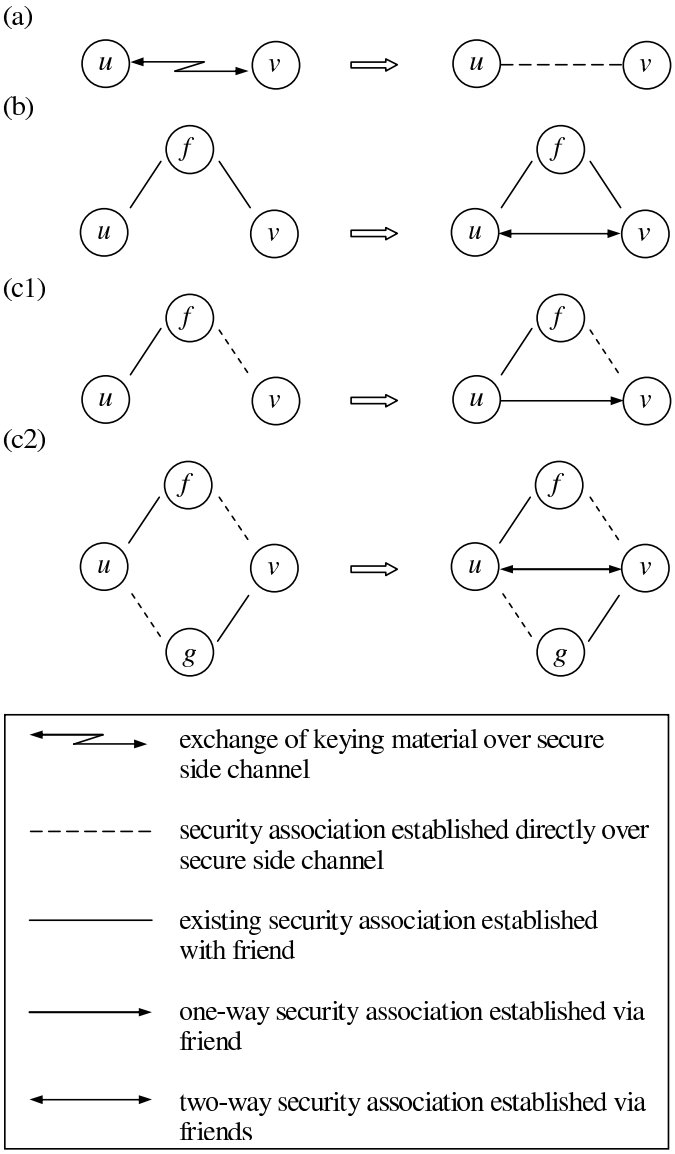

Figure 3.9: Direct and friend-assisted security association establishment [5]

and $match(K_v, a_v)$. In the final two messages users send each other a signature ($\sigma()$) over the radio link. Verifying the signatures with the received public keys serve as proof that u and v knows the corresponding private keys. Note that the triplets and signatures could have been exchanged purely over the side channel, but Protocol 1 minimizes the amount of data sent over the side channel by sending hashes (ξ_u, ξ_v) as an integrity check code over the side channel. This allows u and v to exchange the rest of the information over the radio interface with the possibility of the man-in-the-middle attack eliminated.

Mechanism (b) uses a common friend f to generate and distribute to each node fresh certificates. Since f shares keying material with both u and v, the user can verify the received certificates from f.

Mechanism (c1) enhances key agreement during physical encounters with friend security associations and is simply a combination of Mechanism (a) and (b) detailed above. This mechanism can be used to establish either a one-way or two-way security association between u and v.

Mechanism (c2) will be discussed in the following section as it is more applicable in a symmetric key setting.

3.8.2.2 Symmetric Key Approaches

In a symmetric key setting the three mechanisms illustrated in Figure-3.9 remain applicable in a different context.

With Mechanism (a) the users use the side channel to exchange all the necessary keying material to set up a shared key between them. The side channel in the symmetric key setting must also provide confidentiality in addition to data integrity (see Section-2.3.1). To avoid attack from passive adversaries the users must be cautioned to activate their side channels with no other users within a "secure range" from them.

In Mechanism (b), users have a common friend f, that plays the role of a trusted authority or trusted intermediatry. There are well established protocols that can be used in such a

setup [41].

Mechanism (c2) can be used if u and v do not share a common friend and in case they do not want the trusted third party to know their shared key. A friend of u, named f and a friend of v, named g are used as two separate paths by u and v to exchange key contributions. Protocol 2 [68] [5] presented below explains Mechanism (c2) in more detail:

Protocol 2 : Mechanism (c2)

msg1	$u \rightarrow v$:	f, r_u
msg2	$v \rightarrow u$:	g, r_v
msg3	$u \rightarrow g$:	$u, \{d_{u \rightarrow g}, request, v, k_u, r_v\} k_{ug}$
msg4	$g \rightarrow v$:	$g, \{d_{g \rightarrow v}, reply, u, k_u, r_v\} k_{vg}$
msg3'	$v \rightarrow f$:	$v, \{d_{v \rightarrow f}, request, u, k_v, r_u\} k_{vf}$
msg4'	$f \rightarrow u$:	$f, \{d_{f \rightarrow u}, reply, v, k_v, r_u\} k_{uf}$
	u, v:	$k_{uv} = h(k_u \parallel k_v)$

In Protocol 2 users u and v use messages 1 and 2 to exchange random numbers (r_u, r_v) and the names (f, g) of their friends. In message 3 and 4 and message 3' and 4', u sends k_u to v via g and while v sends k_v to u via f. All the messages are encrypted with a shared symmetric key k_{xy}, where $x \in [u, v]$ and $y \in [g, f]$. In order to avoid ambiguity each message includes the direction d and purpose of the message (*request* or *reply*). Users u and v generate a shared key k_{uv} by taking the hash of the concatenation of their individual contributions.

3.8.3 Discussion and Comments on Mobility-Based Key Management Approaches

The main characteristic of ad hoc networks is the lack of infrastructure. Nodes therefore are responsible for all network functionality of which routing is the most important. In stationary ad hoc networks, nodes may experience frequent link breakages as the traffic in the network increases. Node mobility significantly increases the frequency of these

link breakages. This sporadic connectivity results in a poor availability feature and a high communication overhead for key management schemes that rely on the routing infrastructure. Another reason why relying on the routing infrastructure is infeasible is that any attack on the routing protocol may render the key management scheme insecure. The routing-security interdependence cycle [92] in either case forces the key management scheme to be independent of the routing mechanism. Clearly getting around this problem requires a complete shift with respect to the key management solutions found in conventional wireline networks. The fact that fully self-organized networks do not support any form of trusted authority, not even during off-line initialization, adds a new dimension to this problem. The authors of [68] [5] detach the key management scheme from the routing infrastructure by exploiting user mobility. The mobility characteristic of MANETs, which are widely regarded as a limiting factor, are turned around as an aid to the key establishment mechanisms.

Dependence on mobility (to bring users within a "secure range" in order to use their secure side channels) is the major disadvantage of [68] [5]. The authors themselves identify that it may take some time to set up a sufficient number of security associations. Their simulation results show that, as intuitively expected, the convergence time decreases with an increase in user mobility. The proposal will thus find a strong application as a complementary solution to other key management solutions and is ideally suited to establish security associations on the application layer in a self-organized setting [5].

3.9 Parallel Key Management Approaches

In [6], the authors propose a multiple key management approach by combining a distributed certificate authority (Section-3.6) and certificate chaining (Section-3.3). The proposal known as *composite* key management is based on two fundamental principles. Firstly, key management should be shared between multiple nodes and secondly a trusted third party is required as an anchor of trust. Here certificates, as proposed in [3], are stored and distributed by nodes in a self organized nature. [6] shows how a DCA can be used in *parallel* with certificate chaining to eliminate some of the weaknesses of the

certificate chaining approach. The approach increases availability of the key management service since nodes can use either service to obtain keying material.

The remainder of the discussion on the proposals presented in [6] will focus on the modifications and additional mechanisms added to certificate chaining and the DCA approach.

3.9.1 System Analysis

3.9.1.1 Metrics of Authentication

By introducing *authentication metrics* (confidence values) an attempt is made to provide users with a tool to calculate the level of trust that they can place in an instance of authentication. Users thus assign a confidence value to certificates based on their relationship with the certificate owner. In [6] the confidence value is also extended to incorporate the DCA as a trusted third party.

3.9.1.2 Trust Model

In the certificate chaining approach [3] users form a chain of trust by issuing certificates to other nodes in the network with which they have some relationship or have adequate reason to trust the binding between the node's identity and public key. In [6], the authors illustrate this concept by means of an example: if Alice trusts that Bob is the holder of a public/private pair, Alice issues a certificate containing Bob's ID, Bob's public key and other attributes such as a certificate lifetime parameter. Alice then generates a digital signature on Bob's certificate vouching for the certificate's authenticity.

Similar to [3], composite key management captures trust relationships between nodes in a certificate graph where the edges represent a digital certificate and vertices public keys. The edges are also coupled with a confidence value set by the certificate issuer and assigns a level of trust to the issued certificate.

An example of a certificate chain is given in Figure-3.10 [6]. In the example, if Bob wants

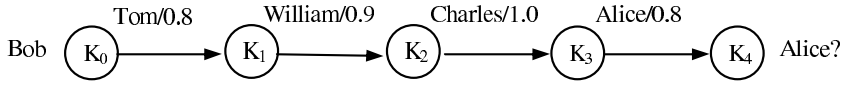

Figure 3.10: Certificate chaining example [6]

to authenticate Alice, Bob needs to calculate a confidence value for the entire chain length. This is done by firstly multiplying all the confidence values for each edge together to form what is called a *raw confidence value*. To get the final confidence value, the chain length d and probability p of the nodes in the chain being compromised, must also be considered. The raw confidence value thus needs to be multiplied by an attenuation factor $(1-p)^{d-1}$ which yields the final confidence value for the chain as a whole. The user utilizes the final confidence value to make a decision on whether to grant the authentication or reject the chain as a possible authentication path.

3.9.1.3 Security Level of DCA

Where certificates in the certification graph are assigned an authentication metric, the DCA is assigned a security level (SL) reflecting the probability that an adversary can compromise the DCA. The security level is calculated as follows:

$$SL = 1.0 - \frac{\binom{n}{k}}{\binom{M}{c}} \tag{3.3}$$

where n is the number of server nodes in the CA, k the *crypto threshold*, M the total number of nodes in the network and c the number of nodes the most powerful adversary can compromise in a fixed time frame.

A system model example is given in Figure-3.11 [6] showing the composition of certificate chaining and a DCA. The certificates are assigned authentication metrics and the DCA a security level, as explained above.

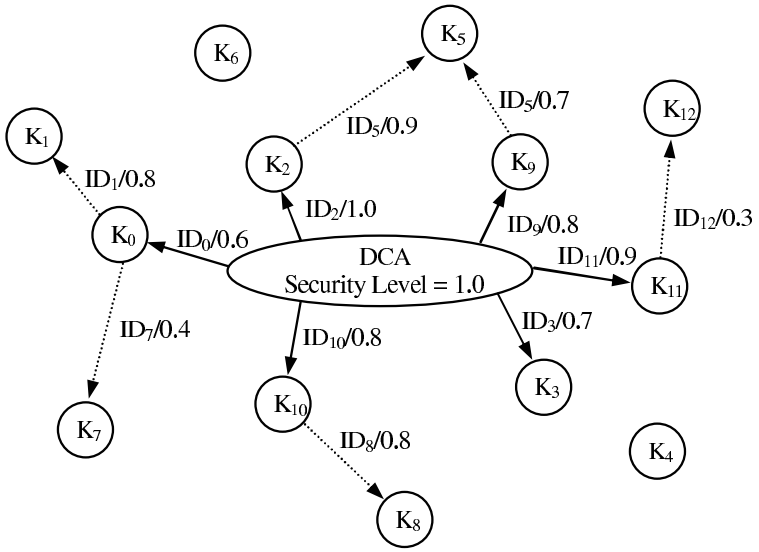

Figure 3.11: Example system model showing DCA composed with 1-hop Certificate Chaining [6]

3.9.2 Discussion and Comments on Parallel Key Management Approaches

The approach of merely combining the distributed certificate authority scheme [1] [65] and certificate chaining scheme [3] fails to adequately address the key management problem in MANETs: the composite key management scheme [6] inherits all the weaknesses of the distributed certificate authority approach as stipulated in Section-3.3.2. In fact, it further degrades the security as it only solves the availability problem of [3] while inheriting its weak authentication property (see Section-3.6.3). The scheme proposed in [3] was designed for "open" MANETs. The authors of [6] claim that they improve on [3], but in fact they are proposing a scheme for an entirely different application.

Combining two key management approaches to eliminate the disadvantages of the other will in most cases not be effective in MANETs: the author has investigated a combination of *all* key management schemes referenced in this thesis. Only the mobility-based approaches (Section-3.8 [68] [5]) can be added as a complement to some schemes to enhance the certificate exchange process by exploiting user mobility. For example, combining the mobility-based approach and the certificate chaining approach will clearly be more effective that combining any one of the two with the distributed certificate authority approach: the disadvantages of the distributed certificate authority approach is, in the view of the

author, unavoidable.

Researchers should not be discouraged from looking at ways to combine the existing key management schemes, but take caution not to create new disadvantages in the process. A complex interaction between the two schemes may also close the window for a strong security argument.

3.10 Conclusions

This chapter presented a survey on peer-to-peer or pairwise key management for mobile ad hoc networks (MANETs). Investigation by the author within the published proposals has shown that the protocols can be grouped into several categories. Each category was discussed by introducing at least the original protocol from within the grouping. This way of categorizing the available protocols gives one the opportunity to establish a deeper insight into the available key management approaches in MANETs and to discuss the strengths and weaknesses of each. From the reviewed key management protocols, the mobility-based approaches [5] are the most feasible for fully self-organized key management on the application layer.

Research by the author within the available publications confirmed that adapting methods originally designed for conventional wireline networks may not be ideal in MANETs. User mobility and the lack of infrastructure result in sporadic connectivity which may result in frequent unavailability of distributed security services. Protocols therefore have to exploit user mobility and nodes within each others transmission range (close proximity) to avoid relying on the routing infrastructure. Dependence on the routing infrastructure during the bootstrapping of the security mechanisms also results in a routing-security interdependence cycle [92].

The remaining obstacles to be eliminated are the dependence on mobility during the bootstraping of the routing security and to minimize user interaction on the application layer. Approaches presented in [105] [106] solve the latter problem by effectively reducing user operation down to the “push of a button”; users will not use security mechanisms

that inconvenience them in anyway.

In the view of the author it is important to decouple key management mechanisms intended for securing application level services from those used to secure the routing infrastructure. A failure in the users' ability to judge the honesty or intent of other users should not jeopardize the security of any basic network service.

The final observation is related to the criteria used by reseachers to analyse key management schemes for MANETs. Key management schemes are designed either for an "open" or "closed" network and consequently aimed at different applications. "Open" or *fully* self-organized MANETs have some inherent security implications (see Section-2.1.3) and must be analyzed accordingly. It is therefore not always possible to compare schemes that assume the existence of a trusted authority with those that are fully self-organized.

Chapter 4

Proposed Peer-to-Peer Key Management Scheme

4.1 Introduction

The main contribution of this thesis is a peer-to-peer key management scheme that is suitable for *fully* self-organized MANETs (see Section-2.1.3). Fully self-organized MANETs will not find application where user access to the network is restricted. The proposed scheme is designed mainly for "open" MANETs where any person with the appropriate equipment can join or leave at random without contacting any trusted authority. "Open" MANETs will thus create commercial applications in a localized geographic context. The main advantage of "open" MANETs is the significantly reduced operating cost for the end-user since "open" MANETs do not rely on an operator to deploy and maintain the network infrastructure. The end-user is required to make only a once-off investment in purchasing an off-the-shelf product such as a handheld computer and the necessary software. Users may form a MANET, for example in a shopping center, which enables them to communicate without paying a network operator for the service. Other possible applications of "open" MANETs are given in Section-2.2.2. It is clear that *fully* self-organized MANETs will, for example, not find application in *vehicular ad hoc networks*, but rather

in networks without strong access control.

Some inherent security implications of an "open" MANET are discussed in Section-2.1.3. The main disadvantage of *fully* self-organized networks is that nodes can adopt as many identities as they have resources to support [12] (see Section-2.1.3 and Section-2.3.3.9). The capability of an attacker with multiple identities in MANETs is an open research problem [5] and not within the scope of this work.

The proposed key management scheme, Self-Organized Peer-to-Peer Key Management (SelfOrgPKM), leverages a variant of the ElGamal type signature scheme, *subordinate public keys* and crypto-based identifiers [110] [45], to achieve low protocol complexity without introducing excessive communication and computational overhead. SelfOrgPKM allows nodes to initialize themselves by generating their own keying material before joining the network. The scheme's operation is fully self-organized, with the burden of key management uniformly distributed between all network participants. Each node is thus its own authority domain which, similar to previous schemes [3] [5], is SelfOrgPKM's main assumption. The nodes establish security associations with their one-hop neighbors on the network layer during route establishment and on the application layer on a need to know basis. With respect to the routing protocol, a bi-directional security association between two nodes, A and B, exists if the nodes have exchanged their tuples [K_A, ID_A] and [K_B, ID_B], where K_i is node P_i's keying material (symmetric or asymmetric keys) and ID_i a unique network identifier/address. Note that K_i and ID_i have to be authentically bound, with the binding between the keying material and node identifier/address publicly verifiable. As with most "open" networks it is left to the user controlled applications to further bind the tuple [K_i, ID_i] to a unique user name. Good examples of application layer key establishment mechanisms can be found in [5] [105] [106].

Subordinate public keys are defined here as public keys that are derived from a user's *self-generated* primary or base public/private key pair. In the thesis the following properties are imposed on subordinate public keys:

1. A valid subordinate public key can only be generated if the entity knows the base/-primary private key.

2. The user can *self-generate* a renewed subordinate public key as frequently as needed.

3. The subordinate private key must be statistically independent of the base private key and other renewed subordinate private keys, i.e. a compromised subordinate private key does not reveal any information about the user's base private key or any future renewed subordinate private keys.

4. There must exist a binding between the user's base public key and subordinate public key that supports non-repudiation.

The chapter is organized as follows: in Section-4.2 the related work is briefly surveyed. Section-4.3 presents a variant on the generalized ElGamal type signatures as a strong cryptographic building block for the proposed subordinate public key generation scheme. Section-4.4 introduces a new subordinate public key generation scheme. In Section-4.5 the new peer-to-peer key management scheme, SelfOrgPKM, for self-organized MANETs is proposed.

This chapter is dedicated to the presentation of the new key management scheme developed by the author. Chapter-5 discusses the security, performance and features of the proposed peer-to-peer key management scheme. The conclusions provided in Section-5.9 provide an overall perspective on the discussions in Chapter-4 and Chapter-5.

4.2 Brief Summary of Directly Related Work

The existing peer-to-peer key management schemes for MANETs were extensively reviewed in Chapter-3. In this section the key management schemes most relevant to self-organized MANETs (some not discussed in Chapter-3) will be briefly summarized.

Capkun *et al.* [3] present a self-organized public key management scheme based on Pretty Good Privacy (PGP) [101]. Similar to PGP, each node disseminates its own certificates and keeps a certificate repository comprising of the certificates of nodes in its local neighborhood. Users share their certificate repositories and mutually authenticate each other's certificate by finding a certificate chain linking their certificates (see Section-3.6).

Montenegro *et al.* [45] and Bobba *et al.* [92] use crypto-based-identifiers to bind node identifiers to public keys. The notion of using partial hashes of a mobile node's public key to form its network address was originally proposed by O'Shea *et al.* [110] as an authentication protocol to protect Mobile IPv6 from address falsification. The crypto-based addresses are used in [92] to protect the basic exchanges between nodes and to bootstrap the routing security mechanism, effectively breaking the routing-security interdependency cycle and solving the address ownership problem.

Lately in [5], Capkun *et al.* have proposed a peer-to-peer key management scheme that relies on user mobility to bring nodes within each other's transmission range, which allows them to exchange their certificates without relying on a secure routing infrastructure. The fully self-organized version of the scheme requires nodes to use a secure side channel between the users' personal devices to authenticate each other and to set up shared session keys. The secret side channel can be a short range connectivity system such as infrared or a physical wire [5] (see Section-3.8).

The impact of these protocols on the development of the proposed key management scheme, SelfOrgPKM, is considered in Chapter-5.

4.3 Modified ElGamal Signature Scheme

In this section a *modified* ElGamal type signature scheme is presented, developed from the generalized ElGamal signature, introduced by Horster *et al.* [111]. The presented ElGamal variant will be used as a strong cryptographic building block for the proposed subordinate public key generation scheme in Section-4.4.

4.3.1 System Parameter Setup

The following system parameters are generated as usual:

p, q two large primes, such that $q \mid (p-1)$.

g generator of the cyclic subgroup of order q in $(Z)_p^*$.

$H(\cdot)$ collision free one-way hash function.

x_P private key of user P.

y_P public key of user P, where $y_P = g^{x_P} \ mod \ p$.

It is noted that p can be chosen to ensure a known factorization of $p-1 = p_1^{e_1}, \ldots, p_k^{e_k}$, where $p_1, \ldots, p_k$ are distinct primes. The value g can be chosen as follows [41]:

1. Choose a random value $h \in (Z)_p^*$.
2. Check if $h^{\frac{(p-1)}{q}} \not\equiv 1 \ mod \ p$
3. If the incongruence holds then $g = h^{\frac{(p-1)}{q}}$ is a valid primitive element, otherwise repeat from (1) until the chosen value g from $(Z)_p^*$ satisfies the incongruence.

4.3.2 Signature Generation

User P selects a random number $k \in [1, q-1]$ and computes a public commitment r as:

$$r = g^k \ mod \ p \tag{4.1}$$

User P signs an arbitrary message m by solving the following congruence:

$$s \equiv x_P + [H(m \parallel r)]k \ mod \ q \tag{4.2}$$

The set (s, r) is the signature of user P on message m.

4.3.3 Signature Verification

Any outsider can use user P's public key y_P to verify the validity of the signature (s, r) for a message m by checking whether the following equation holds:

$$g^s = y_P r^{H(m\|r)} \; mod \; p \tag{4.3}$$

4.4 Proposed Subordinate Public Key Generation Scheme

The proposed subordinate public key generation scheme is based on the modified ElGamal signature variant presented in Section-4.3 and borrows concepts from *parameter hidden* signature schemes [112] [113].

Self-certified public keys [114] also use concepts from parameter hidden signature schemes. The notion of self-certified public keys was first introduced in [114] and later extended in [100]. In self-certified public key schemes, the public key is computed by both the trusted authority and the user, without the authority gaining knowledge of the corresponding private key.

Consider at first the existing scheme given in [100] that can be summarized as follows:

1. A certificate authority (CA) chooses random number $k'_A \in_R Z_q^*$, computes $r'_A = g^{k'_A}$ and transmits r'_A to party A.

2. A chooses random number $a \in_R Z_q^*$, computes $r_A = r'_A g^a$ and transmits (ID_A, r_A) to CA.

3. CA computes the signature parameter, $s'_A = x_{CA}[h(ID_A, r_A)] + k'_A$ and sends s'_A to party A.

4. Party A computes the private key $x_A = s'_A + a$. The tuple (r_A, x_A) can be seen as the signature of the CA on A's identity, ID_A. Party A verifies the signature of the

CA and calculate the corresponding public key using equation (4.4):

$$y_A \equiv g^{x_A} = y_{CA}^{h(ID_A, r_A)} \cdot r_A \tag{4.4}$$

The proposed scheme, given below, differs from the above existing scheme in a few instances: 1) party A becomes its own authority domain, hence their is no CA involved; party A will thus play the role of the CA, 2) the proposed scheme is based on the modified ElGamal signature scheme given in Section-4.3, in contrast to the blind Schnorr signature scheme introduced in [113] and 3) the scheme presented in [100] addresses a completely different application.

The proposed subordinate public key scheme is as follows:

The system parameters introduced in Section-4.3.1 are applicable. It is assumed that party A has generated its own public key/private key pair as follows: party A chooses a random number $x_A \in_R [1, q-1]$ as its private key and computes its corresponding public key as $y_A = g^{x_A} \ mod \ p$.

Party A can generate a subordinate public key from its base key pair (x_A, y_A) that satisfies the properties defined in Section-4.1 as follows:

- Party A chooses a random number $k_A \in_R [1, q-1]$ and computes $r_A = g^{k_A} \ mod \ p$.
- Party A computes its new subordinate private key as:

$$x'_A = x_A + [H(KI_A)]k_A \ mod \ q, \tag{4.5}$$

  where the subordinate key information is defined as $KI_A = [ID_A \parallel y_A \parallel r_A \parallel SerNo \parallel IssueDate \parallel ValPeriod \parallel ExtInfo]$. Note that the contents of KI_A can be altered based on the network policy, where ID_A is the identity of party A, $SerNo$ a unique sequence number, $IssueDate$ the date of issuing the certificate, $ValPeriod$ the validity period and $ExtInfo$ some additional extension information.

- Finally party A computes its corresponding subordinate public key as:

$$y'_A = g^{x'_A} = y_A(r_A)^{H(KI_A)} \; mod \; p \tag{4.6}$$

Party A can renew its subordinate key pair with a self-organized subordinate key renewal procedure: party A simply chooses a new random number $k'_A \in_R [1, q-1]$ and computes its renewed subordinate private key as:

$$x''_A = x_A + [H(KI'_A)]k'_A \; mod \; q, \tag{4.7}$$

where $KI'_A = [ID_A \parallel y_A \parallel r'_A \parallel SerNo + 1 \parallel IssueDate' \parallel ValPeriod' \parallel ExtInfo']$.

4.5 Proposed Fully Self-Organized Peer-to-Peer Key Management Scheme

The discussion on the proposed peer-to-peer key management scheme is started with a problem statement. This may also be seen as a design specification. In Section-5.7, it will argued that the proposed scheme satisfies all these requirements.

4.5.1 Problem Statement

The problem statement will be: the design, analysis and performance evaluation of a peer-to-peer key management scheme for *fully* self-organized mobile ad hoc networks. The characteristics of *fully* self-organized mobile ad hoc networks are given in Section-2.1.

The proposed key management scheme must satisfy the following general requirements (see Section-3.1.2):

1. The key management scheme must support a hybrid (symmetric and asymmetric) key management system. In order to relax the requirement for a confidential channel, the initial keying material must be distributed via an asymmetric key system.

2. All keying material (public keys) must be publicly verifiable to be *explicitly* authentic.

3. The *key freshness* property in Section-3.1.2 is adapted for an asymmetric system, hence it is required that all users' private keys are independent. A user's renewed keying material must also be independent.

4. The scheme must protect the private keys of users in the presence of the active, insider adversary, thus also provide resistance to known key attacks and ensure forward and backward secrecy.

5. The key management scheme must satisfy the survivability property, hence the scheme must remain available in the presence of threats and failures. Note that unavailability as a result of an attack on the routing and other basic mechanisms, is not within the scope of this work. The key management scheme will have to rely on various other mechanisms, such as an intrusion detection mechanism [115], to fully deal with *The Three Rs* of survivability given in Section-3.1.2.

6. Robustness of the key management scheme must be ensured.

7. All operations of the scheme should optimize the use of communication, storage and computational resources, both in the initialization and post-initialization phases. The scheme should therefore not degrade the network performance.

8. The last requirement to be satisfied as given in Section-3.1.2 is scalability: the key management service must allow for changes in node density.

Although SelfOrgPKM does not explicitly define a symmetric key establishment protocol to set up session keys between users, it is noted that such mechanism should not violate the perfect forward secrecy property.

In addition to the above generic requirements, the following properties must be provided by the proposed key management scheme:

9. The exchange of keying material should be on-demand: each user will create its own keying material and only exchange keying material with other nodes if the

keying material is needed by other mechanisms (such as the routing protocol and user applications).

10. The key management scheme must break the routing-security interdependence cycle as defined in [92]. In fact, it is required that the key management scheme should not be dependent on the routing infrastructure at all.

11. In an "open" network, adversaries should not be able to spoof the network addresses of other nodes. The proposed scheme must thus solve the address ownership problem as defined in [110] [45].

12. The key management scheme must be *fully* distributed. This means than an adversary can compromise any number of nodes and not break the key management service. This property will also preserve the symmetric relationship between the nodes and help to ensure survivability.

13. An increase in node mobility should not affect the operation of the proposed key management scheme. In fact it is required that the key management scheme exploits mobility.

14. Although the key management scheme may exploit mobility it must not rely on node mobility.

15. The setup of security associations should not suffer from any noticeable time delay.

16. A strong security arguement for the proposed key management scheme must be traceable to a hard mathematical problem within a widely accepted security model.

17. Complexity is the biggest enemy of any security system [116]. Low complexity is one of the key design constraints of the proposed key management scheme. The scheme must be practical and thus easy to implement.

18. The scheme should be fully modularized, hence its components must be independent and usable in other applications.

4.5.2 System Model

The proposed peer-to-peer key management scheme for MANETs, called SelfOrgPKM, uses subordinate public keys (presented in Section-4.4) and crypto-based identifiers [110] [45] as strong cryptographic building blocks to set up security associations between nodes. The bootstrapping of the security service introduces minimal communication and computational overhead and does not require any form of off-line or on-line TTP. This property is consistent with the characteristics of fully self-organized MANETs as defined by [3] [5].

The proposed scheme considers a network of wireless nodes with low to high mobility speeds ($1m/s - 20m/s$). The medium access control (MAC) and routing mechanisms are assumed to be generic. The network is open for any user to join or leave at random without restricted access. The specific application of the scheme will therefore not be for military type mobile ad hoc networks, which have a high security demand, but rather for commercial or other less access constrained environments.

It is assumed that there is no pre-existing infrastructure and no form of on-line or off-line trusted authority; before users join the network they have to determine what universal set of system parameters are used in the network. SelfOrgPKM assumes that the users have system parameter sets pre-imaged on their nodes and that users publicly establish which set to use. Secure system parameter exchanges, without prior knowledge or any user interaction, is an interesting direction for future research. Each node generates a discrete logarithm public/private key pair. By hashing the base public key with a one-way collision resistant hash function, nodes obtain a unique network identifier/address. The proposed scheme inherits this idea from the original proposal of O'Shea *et al.* [110]. The base or originally generated key pair is never used for any real communication to protect it from attacks on the cryptographic algorithms used to secure communication [100] [41]. The users rather derive a second or subordinate public/private key pair from their base key pair as specified in Section-4.4. Each user then generates a self-certificate to bind other useful information to their keying material such as a unique sequence number, expiry date etc.

Any user with a valid self-generated certificate can join the network. As in the initialization

phase, each node is its own authority domain during network operation and therefore responsible for the renewal and dissemination of its own self-certificates. Nodes within each other's transmission range exchange certificates during route establishment on the network layer and users exchange their certificates on the application layer on a need to know basis, thus only when they want to communicate securely.

4.5.3 Adversary Model

SelfOrgPKM considers a straightforward, general adversary model. An adversary is a malicious node that uses every means available to break the proposed key management scheme. Any *active* adversary can eavesdrop on all the communication between nodes, modify the content of messages and inject them back into the wireless channel. When a node is compromised all its public and private information is exposed to the adversary. As detailed in Section-2.3.2, an active, *insider* adversary is assumed.

The operation of SelfOrgPKM is divided into a node initialization phase which is executed by each node before the node joins the network and post-initialization which executes during network operation.

4.5.4 Initialization Phase of SelfOrgPKM

Each node P_i, for $(1 \leq i \leq n)$, creates a base public/private key pair (x_i, y_i) by choosing a random number $x_i \in_R [1, q-1]$ as its base private key and computes its corresponding public key as $y_i = g^{x_i} \; mod \; p$. It is assumed that each node has an authentic image of the system parameters, as specified in Section-4.3.1.

Each node generates a unique identifier (ID_i) that is bound to its base public key y_i as follows:

$$ID_i = H(y_i) \tag{4.8}$$

SelfOrgPKM requires ID_i to be used as the node's network address or as a fixed part of the address. Note that this requirement places no constraint on the structure of the network

addresses: the entire hash output, ID_i, can be used in MANETs with flat, static addresses or only a part of the output can be used in MANETs with dynamic addressing. Note that SelfOrgPKM can easily be extended to incorporate strong network access control at the cost of losing the self-organized feature: an off-line trusted authority can bind ID_i, y_i and a unique username by generating a certificate for P_i signed by the authority.

Each node P_i uses its base public/private key pair (x_i, y_i) to generate a subordinate public/private key pair (x'_i, y'_i) as specified in Section-4.4.

Note that P_i's base key pair (x_i, y_i) is never used for real communication. Rather, each P_i uses its subordinate key pair (x'_i, y'_i) for securing actual communication.

To obtain an explicitly authentic key pair each node uses its newly obtained subordinate private key x'_A to sign the key information content, KI_i (concatenated with its subordinate public key y'_i and public commitment β'_i) via the modified ElGamal signature scheme presented in Section-4.3. Node P_i's self-certificate can then be defined as: $SelfCert'_i = [KI_i \parallel y'_i \parallel \alpha'_i \parallel \beta'_i]$, where (α'_i, β'_i) is the appended signature on $KI_i \parallel y'_i \parallel \beta'_i$.

4.5.5 Post-Initialization Phase of SelfOrgPKM

The post-initialization phase commences after network formation. Each node must perform the initialization phase, as presented in Section-4.5.4, before joining the network.

4.5.5.1 Certificate Exchange and Authentication

Certificate exchange takes place between nodes on a peer-to-peer, need-to-know basis. Nodes set up a bidirectional security association by exchanging their renewed self-certificates, $SelfCert'$. SelfOrgPKM requires all nodes to exchange self-certificates with their one-hop neighbors on the network layer. As shown in Figure-4.1, nodes within each other's transmission range exchange their certificates during route establishment. The scope of this work is limited to on-demand routing without binding the proposed scheme to a specific routing protocol. The source node starts as usual with a broadcast route request. Two

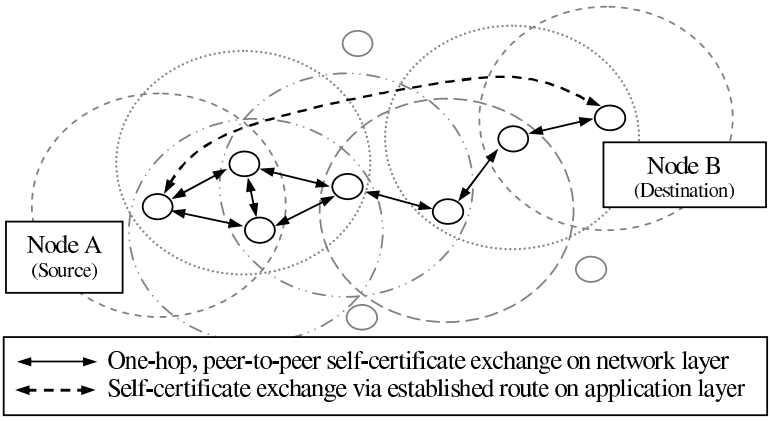

Figure 4.1: SelfOrgPKM certificate exchange

unicast messages are needed for subsequent certificate exchange (if the source and neighboring node have not done so already): one message from the neighboring node and one message from the source node. This process continues until the route request reaches the destination node, hence the route requests serve as triggers for certificate exchanges between neighboring nodes in the route discovery phase.

It is trivial to see that the one-hop network layer certificate exchange mechanism makes the security scheme independent of the routing-security interdependence cycle as defined in [92].

The example, illustrated in Figure-4.1, also explains certificate exchanges at the application layer: assume $Node_A$ wants to communicate securely with $Node_B$. In the first round $Node_A$ sends to $Node_B$ a $CertRequest$ requesting $SelfCert'_B$ from $Node_B$ over the established route. Note that $CertRequest$ contains the certificate $SelfCert'_A$. If $Node_B$ grants the request it replies in the second round with $SelfCert'_B$. Note that this two-round procedure requires no synchrony between $Node_A$ and $Node_B$.

Refer to Section-5.5 for a detailed example of SelfOrgPKM's certificate exchange mechanism.

The self-certificates and subordinate public keys of $Node_A$ and $Node_B$ are authenticated as follows:

1) Each node implicitly authenticates the base public key of its peer node by checking if

Equation-4.8 holds.

2) Next, the peer nodes implicitly authenticate the subordinate public keys of their peers by checking if Equation-4.6 holds.

3) Finally each node validates the self-certificate of its peer node, $SelfCert'_i$, by verifying the signature (α'_i, β'_i) on $[KI_i \parallel y'_i]$. This explicitly authenticates both the base public key y_i of $Node_i$ and the subordinate public key y'_i. If the node identifier ID_i matches the hash output the node is also assured that the identifier/address has not been spoofed by $Node_i$.

If all three of the above verification steps hold, then the subordinate public key y'_i is explicitly authentic and securely bound to the base public key y_i, which in turn is securely bound to the nodes *statistically* unique identifier/address.

If the certificate transfer fails due to error-prone connectivity or unsuccessful authentication, $Node_A$ will retry by resending $CertRequest$.

For efficiency reasons nodes can use symmetric key schemes to secure all subsequent messages. This can easily be achieved by using the available authenticated public keys to establish a session key between peers.

4.5.5.2 Certificate Revocation

SelfOrgPKM makes use of a self-revocation system based on a self-organized subordinate key renewal procedure. As mentioned in Section-4.5.4, nodes do not use their base key pair for any real communication, but must derive a subordinate key pair (x'_i, y'_i) from the base key pair which is then used for actual communication. This significantly reduces the chance of a successful attack on a node's base key pair [100] [41]. The self-organized key renewal process given in Section-4.4 can be used by the node to obtain a renewed key pair (x''_i, y''_i) at any point in time during the post initialization operation of the network. The node will thus derive a new private key $x''_i = x_i + H(KI'_i)k'_i \ mod \ q$ and generate a new self-certificate $SelfCert''_i = [KI'_i \parallel y''_i \parallel \alpha''_i \parallel \beta''_i]$. The renewed certificate $SelfCert''_i$ can be sent to the node's frequent contacts or offered to other nodes on communication

initialization. Since nodes are responsible for their own keying material they can renew their subordinate key pair as frequently as desired. Nodes will however most likely renew their key pair in two instances: when they suspect that their subordinate private key has been compromised or when their set validity periods $ValPeriod'$ have expired.

4.6 Summary and Conclusion

This chapter proposed a novel peer-to-peer key management scheme for fully self-organized MANETs, called SelfOrgPKM. The introduction of the proposal was started by briefly summarizing the existing literature closely related to SelfOrgPKM (Section-4.2). Two generic building blocks for SelfOrgPKM was presented. The first building block, introduced in Section-4.3, is a signature scheme developed from the generalized ElGamal signature variant [111]. The ElGamal signature forms the basis of the second building block of SelfOrgPKM. The novel notion of subordinate public keys was introduced. Furthermore, a subordinate public key generation scheme was proposed that allows a user to derive a second public/private key pair from its base key pair (Section-4.4).

The discussion on SelfOrgPKM started in Section-4.5.1 with a problem statement that clearly defines the design specifications for the proposed key management scheme. The following sections introduced the system and adversary models for SelfOrgPKM (Section-4.5.2 and Section-4.5.3). This laid the foundation to present the details of SelfOrgPKM in Section-4.5.4 and Section-4.5.5. The discussion was divided into an off-line initialization phase and on-line post-initialization phase. The initialization phase describes how users generate their own keying material (self-certificates) prior to joining the network. The post-initialization phase deal primarily with the novel certificate exchange mechanism on the network layer.

The discussion on the proposed scheme is continued in Chapter-5. Chapter-5 focuses on the features, security and performance evaluation of SelfOrgPKM and its building blocks. It is also argued that SelfOrgPKM satisfies all the design specifications given in Section-4.5.1.

Chapter 5

Performance and Security Evaluation of the Proposed Peer-to-Peer Key Management Scheme

5.1 Introduction

The proposed peer-to-peer key management scheme for MANETs, presented in Section-4.5, makes use of subordinate public keys and crypto-based identifiers as building blocks to effectively eliminate the need for any form of off-line or on-line TTP. Self-organized MANETs by definition do not have any form of off-line TTP [3] [5] (see Section-2.1.3). This makes schemes such as [1] [2] [65] unsuitable for self-organized MANETs. The weaknesses of these existing schemes extend into network formation. They use a distributed certificate authority (DCA) as an on-line TTP which can be attacked. The proposed scheme avoids these weaknesses by using a fully distributed system where each node becomes its own authority domain.

The existing schemes that take the characteristics of fully self-organized MANETs into consideration have the following main weaknesses (see Section-4.2):

1) The certificate chaining based key management approach, presented in [3], only provides weak certificate authentication and may fail to provide certificate chains between all node pairs in the network during the initialization phase (see Section-3.6.3).

2) The major weakness of the crypto-based identifier approach [110] [45] [92] is that users cannot revoke their public keys without changing their network addresses and/or identifiers [5]. In [92], the routing protocol control packets are extended by appending a users public key to route requests. With the high number of route requests sent by the on-demand routing protocols this leads to significant additional overhead.

3) The mobility based key management approach in [5] introduces a time delay in the setting up of the security associations. The fully self-organized scheme relies on node mobility to ensure that sufficient nodes come into physical contact. Asking of users to actively set up security associations with every node they meet may be a strong requirement. As noted by the authors of [5], the fully self-organized methods proposed in [5] are ideally suited to set up security associations on the application layer (see Section-3.8.3).

In general, all three of these approaches have strong points which are adequately highlighted by the authors. Some of these concepts have been incorporated into SelfOrgPKM and clearly referenced when used.

The proposed scheme does not suffer from any of the above weaknesses: SelfOrgPKM inherits the main benefits of crypto-based identifiers [110] [45] [92] as a means of solving the address ownership problem. Subordinate public keys introduced in Section-4.4 are used for real communication, leaving an adversary with a brute-force attack as the only option to compromise a node's base public/private key pair and/or identifier [100] [41]. The subordinate key pairs can be easily renewed without having to modify the base public key pair which keeps the nodes' identifiers constant. With the proposed certificate exchange mechanism users do not experience any noticeable time delay in the set up of security associations and distribution of their own certificates. SelfOrgPKM exploits the routing

control packets to trigger localized certificate exchanges without extending the routing packets. Certificates can be explicitly authenticated with low computational overhead.

The characteristics of "open" networks inherently make them vulnerable to Sybil attacks, as defined in [12] (see Section-2.1.3). As mentioned before, the effect of nodes with multiple identities in MANETs could be an interesting future research topic. More importantly, in "open" networks, nodes should not be able to impersonate other network nodes, hence the key management protocol must ensure that a node's identity/address cannot be falsified and that the binding between the node's identity and keying material cannot be forged. The user's certificate must therefore be explicitly authenticated. This will be the focus in the remainder of the security discussion.

5.2 On the Security of SelfOrgPKM

Considering the proposed scheme's *system model* and *adversary model*, given in Section-4.5.2 and Section-4.5.3 respectively, it becomes clear that an adversary will attack the scheme mainly in the following ways:

1. SelfOrgPKM uses a one-way collision resistant hash function to uniquely bind the nodes' public keys and network identifiers. From Equation-4.8, an adversary can attempt to spoof the identity/address of user P_i, by generating a forged public key y_i' that hashes to the same ID_i. This is analogous to finding a collision on a strong hash function such that $ID_i = H(y_i) = H(y_i')$.

2. The proposed key management scheme uses the subordinate public key generation scheme given in Section-4.4 with a twofold objective. Firstly, the scheme allows users to derive a secondary public/private key pair used for real communication thereby forcing the adversary to directly derive the private key from the public key, i.e. find $log_g\ y_i$. Secondly, it provides the nodes with an efficient computational and communication method of frequently renewing their keying material. Thus, from

Equation-4.6, an adversary can attempt to obtain a forged subordinate public key y_i'' that satisfies $y_i'' = g^{x_i''} = y_i(r_i')^{H(KI_i')} \; mod \; p$.

3. Users generate self-certificates to bind their key pairs to their keying information KI_i as defined in Section-4.4. An adversary may attempt to forge a self-certificate by binding a forged subordinate public key y_i'' to the users valid keying information KI_i.

4. The only messages an adversary can intercept and alter are the certificates exchanged between nodes. While SelfOrgPKM prevents adversaries from forging certificates, mitigating all attacks on the network and lower layers that may thwart reliable communication, is not within the scope of this work. SelfOrgPKM does however provide the network layer with cryptographic keying material without relying on a route establishment mechanism, thereby breaking the routing-security interdependency cycle [92].

The goal of the discussion on the security of SelfOrgPKM is to show that the proposed scheme is secure within the given informal *adversary model*, i.e. it avoids the main angles of attack pointed out in (1 to 4) above. The strong modularization of SelfOrgPKM makes it possible to break up the angles of attack as described above. The security of the self-certificate generation and subordinate public key generation procedures are largely based on the security of the ElGamal type signature scheme presented in Section-4.3. The security discussion is thus divided into three parts:

- Firstly, the modified generalized ElGamal type signature variant, presented in Section-4.3, is proved secure within the combined security model, the Random Oracle and Generic Model (ROM+GM), proposed by Schnorr *et al.* [117] [118]. Although this model is mainly of theoretical interest, the proof provides some form of guarantee that the signature scheme cannot be broken based on widely accepted cryptographic assumptions. The two main assumptions in ROM+GM are the existence of an ideal hash function and ideal group G of prime order q. There exist hash functions and groups that have shown to be practical (but limited) instances of such ideal oracles.

- Secondly, it is shown that verifying a user's self-certificate via verification of the ElGamal type signature on the self-certificate content $KI \parallel y'$ and validating the subordinate public key with Equation-4.6, yields the user's subordinate and base public key explicitly authentic.
- Lastly, the probability of an adversary finding a collision on a strong hash function and thus spoofing network addresses or identifiers is investigated.

The philosophy behind the given security analysis is supported by Koblitz and Menezes in [119].

5.2.1 Security Proof for the Presented ElGamal Signature Scheme

In the following proof reference is made to the combined security model, the Random Oracle and Generic Model (ROM+GM), proposed by Schnorr *et al.* [117] [118]. Note that the formal adversary model as defined by Schnorr *et al.* is applicable.

In the first part of the security proof for the proposed peer-to-peer key management scheme, SelfOrgPKM, it will be shown that the modified generalized ElGamal type signature variant presented in Section-4.3, is secure against the *one-more signature forgery* attack [117] in the ROM+GM security model.

Theorem 1. *Let a generic adversary $\mathcal{A}$ interact with a signer and be given the generator g, the public key y and an oracle for H. $\mathcal{A}$ performs t generic steps which include l sequential signer interactions. With a probability space consisting of y, H and coin flips of the signer, it is not possible for $\mathcal{A}$ to produce $l+1$ signatures with a probability better than $\frac{\binom{t}{2}}{q}$.*

In the following proof *Lemma 1* and *Lemma 2* are those defined and proved in [117].

Proof. [following Schnorr *et al.* [117]]

As given by *Lemma A* defined below, the group element $f_{i'} = g^{\frac{s'_i}{c'_i}} g^{-\frac{x}{c'_i}} = g^{\langle \alpha_{\mathbf{i}'}, (1,x,\mathbf{k}) \rangle}$ for an arbitrary $i \leq t'$. $\mathcal{A}$ receives hash query $c'_i = H(m \parallel g^{\frac{s'_i}{c'_i}} g^{-\frac{x}{c'_i}})$ and needs to find s'_i

which satisfies Equation-5.3. The adversary $\mathcal{A}$ is thus required to solve a linear polynomial $x+c_i'\langle\alpha_{\mathbf{i}'},(1,x,\mathbf{k})\rangle$ at $(x,\mathbf{k})$. By *Lemma 2* presented in [117], x is statistically independent from $(\alpha_{\mathbf{i}'},(1,x,\mathbf{k}))$, excluding prior collisions $f_j = f_k$. By *Lemma 1* presented in [117], it is known that such collisions will only occur with an upper bound probability of $\frac{\binom{t'}{2}}{q}$. On the other hand, by *Lemma A*, adversary $\mathcal{A}$ must choose $c_1,\ldots,c_l$ for each signature (m_i',c_i',s_i') that satisfies Equation-5.1 such that x cancels out. In the case of a sequential attack, without any collisions among the computed group elements $f_1,\ldots,f_{t'}$, the system of $l+1$ equations for $c_1,\ldots,c_l$ is solvable with an upper bound probability of $\frac{\binom{t''}{2}}{q}$, where t'' denotes the number of queries to H [117]. It follows from $\frac{\binom{t'}{2}}{q}+\frac{\binom{t''}{2}}{q}\leq\frac{\binom{t}{2}}{q}$, that $\frac{\binom{t}{2}}{q}$ is the highest probability for $\mathcal{A}$ to succeed in a sequential, *one-more signature* attack on the signature scheme presented in Section-4.3. □

Lemma A. *Let the triplet (m_i',c_i',s_i') be a signature with a probability better than $\frac{1}{q}$. The c_i'-coordinate then coincides with the value $H(m \parallel f)$ corresponding to the hash query $(m \parallel f)$. From Equation-4.3, $g^k = g^{\frac{s}{c}}g^{-\frac{x}{c}}$. The hash query $(m \parallel f) \in G \times M$, satisfies $c_i' = H(m \parallel f) = H(m \parallel g^{\frac{s_i'}{c_i'}}g^{-\frac{x}{c_i'}})$, where the group element $f = f_i'$ for some arbitrary $1 \leq i' \leq t'$. The parameters (m_i',c_i',s_i') also satisfy:*

$$c_i' = \frac{1}{-\alpha_{i',1}+\sum_{k=1}^{l}\left[\alpha_{i',k}\,c_k^{-1}\right]} \tag{5.1}$$

$$s_i' = c_i'\left[\alpha_{i',0}+\sum_{k=1}^{l}\alpha_{i',k}\frac{s_k}{c_k}\right] \tag{5.2}$$

In the following proof, *Lemma 2* is as defined and proved in [117].

Proof. [following Schnorr *et al.* [117]]

Since $1 \leq i' \leq t'$ denotes the index of f among the computed group elements $f_1,\ldots,f_t'$, the group element can be written as $f_{i'} = g^{\frac{s_i'}{c_i'}}g^{-\frac{x}{c_i'}} = g^{\langle\alpha_{\mathbf{i}'},(1,x,\mathbf{k})\rangle}$. It follows from the previous equation and $k_k = \frac{s_k}{c_k}-\frac{x}{c_k}$, that:

$$s_i' = x + c_i'\log_g\left[g^{\frac{s_i'}{c_i'}}g^{-\frac{x}{c_i'}}\right] = x + c_i'\langle\alpha_{\mathbf{i}'},(1,x,\mathbf{k})\rangle \tag{5.3}$$

$$s'_i = c'_i \left[\alpha_{i',0} + \sum_{k=1}^{l} \alpha_{i',k} \frac{s_k}{c_k} \right] + x \left[1 + c'_i \left[\alpha_{i',1} - \sum_{k=1}^{l} \alpha_{i',k} \frac{1}{c_k} \right] \right] \quad (5.4)$$

In order for the generic adversary $\mathcal{A}$ to calculate the correct s'_i, $\mathcal{A}$ must find c'_i such that x cancels out. $\mathcal{A}$ must therefore select $c_1, \ldots, c_l$ that satisfy Equation-5.1.

If x cancels out, s'_i can be computed by $\mathcal{A}$ as specified by Equation-5.2.

In the case that x does not cancel out in the equality given by Equation-5.4, the equality will only hold with probability $\frac{1}{q}$ since x is statistically independent of non-group data by *Lemma 2* presented in [117]. □

5.2.2 On the Security of the Proposed Subordinate Public Key Generation Scheme

Subsequent security discussions will take on a more informal approach. From any entity's perspective Equation-4.6 can only provide *implicit* authentication of subordinate public key y'_i, i.e. the verification procedure gives no assurance that P_i knows the corresponding private key x'_i. The authenticity of the subordinate public key only becomes explicit when P_i uses it for a cryptographic procedure which inherently provides a proof of knowledge of x'_i.

An adversary $\mathcal{A}$ that wants to produce a forged subordinate public key must compute a public key $y'_{\mathcal{A}}$ that satisfies:

$$y'_{\mathcal{A}} = y_i \cdot (r_{\mathcal{A}})^{H(KI_{\mathcal{A}})} \; mod \; p \quad (5.5)$$

$\mathcal{A}$ does not know $\log_g y'_{\mathcal{A}}$ and will consequently fail to produce a valid signature that satisfies Equation-4.3. This serves as motivation for introducing self-certificate generation in Section-4.5.4, which allows the subordinate public keys to be explicitly authenticated.

It will thus be appropriate to assess the security of the proposed subordinate public key generation protocol in conjunction with the signature (α'_i, β'_i) on $m_i = [KI_i \parallel y'_i]$ as described in Section-4.5.4. It is noted that (α'_i, β'_i) is produced via the proposed signature scheme presented in Section-4.3. The verification equation on (α'_i, β'_i) is given as:

$$g^{\alpha'_i} = y'_i \cdot (\beta'_i)^{H(m_i \| \beta'_i)} \; mod \; p \tag{5.6}$$

Substituting Equation-4.6 into Equation-5.6 yields:

$$g^{\alpha'_i} = y_i \cdot (r_i)^{H(KI_i)} \cdot (\beta'_i)^{H(m_i \| \beta'_i)} \; mod \; p, \tag{5.7}$$

which has the following signature equation:

$$\begin{aligned} \alpha'_i \;\; &= \;\; x_i + H(KI_i)(k_i) \\ &\quad + H(m_i \parallel \beta'_i)(\log_g \beta'_i) \; mod \; q \end{aligned} \tag{5.8}$$

An entity which explicitly authenticates y'_A via Equation-4.6 and Equation-5.6, indirectly verifies Equation-5.8 in two steps. From Equation-5.7 and Equation-5.8 it is concluded that an adversary $\mathcal{A}$ can only generate a forged subordinate public key with an upper bound probability of $\frac{\binom{t}{2}}{q}$ in the ROM+GM security model (by *Theorem 1*). Furthermore it shows that the verifier of Equation-4.6 and Equation-5.6 can be assured that the party with ID_i, generated via Equation-4.8, knows the base private key x_i corresponding to y_i.

Another point of concern is that a compromise of the subordinate private keys may reveal information about the base or primary private key. From Equation-4.2 and Equation-4.5 it can be seen that the base private key x_i is blinded from the subordinate private key x'_i by the addition of a random number k_i. This is the same mechanism that is used by all ElGamal type signatures to protect private keys from being derived from valid signatures. An adversary $\mathcal{A}$ that compromises a subordinate key pair (x'_i, y'_i) therefore has the same probability of gaining knowledge of the base private key x_i as someone with a valid signature (s_i, r_i), generated via the signature scheme presented in Section-4.3.

Note that the subordinate public key generation scheme, proposed in Section-4.4, satisfies all the requirements of a subordinate scheme as defined in Section-4.1.

5.2.3 On the Security of Hash Based Identifiers

The security of crypto-based identifiers has been extensively reviewed in [45] [120] [110] [92]. To avoid repetition this section will only briefly consider the security of hash functions. As noted from the specification of the proposed key management scheme, SelfOrgPKM is not bound to any specific hash function. Such a secure hash function can be carefully chosen on deployment of the protocol. For example, currently SHA-1 [121] with a 160-bit output will provide adequate security since 2^{80} hash operations are needed to find a collision on SHA-1 using a brute-force attack. The most recent known attack on SHA-1, presented in [122], shows that a collision on SHA-1 can be found with a complexity of less than 2^{69} hash operations. Note however, that an attacker also has the additional computational overhead of one full exponentiation in order to find a valid public key before computing the hash function. Clearly the additional time complexity of the exponentiation makes the spoofing of network addresses impractical.

5.3 On the Efficiency of SelfOrgPKM

SelfOrgPKM is fully distributed, preserving the symmetric relationship between nodes as required in MANETs. The proposed peer-to-peer key management scheme thus places the same communication and computational burden on each node, which in the view of the author, is the first step towards mitigating selfishness attacks and extending the nodes' battery life. In the next two subsections the performance of SelfOrgPKM will be investigated in an ideal network setting, hence assuming guaranteed connectivity. Section-5.5 evaluates the performance of SelfOrgPKM in a simulation study where factors such as connectivity and route failures (due to the error-prone wireless channel, node mobility etc.) have an impact on the system operation.

5.3.1 Efficiency of SelfOrgPKM Initialization Phase

The initialization phase is performed by each node before joining the network and therefore has no impact on network performance. This process should however still be as efficient as possible. Each node P_i performs 4 exponentiations (exp), 3 random number generations (R_{gen}) and 3 hash computations ($H(\cdot)$) (The 2 multiplications and 2 summations have insignificant impact on the time complexity in comparison with the exponentiations). The initialization phase has no communication cost.

5.3.2 Efficiency of SelfOrgPKM Post-Initialization Phase

The on-line post-initialization phase of SelfOrgPKM results in little overhead for each node. A node renewing its self-certificate has to perform only 2 signature generations and 1 exponentiation to compute its renewed subordinate public key with a total cost of 3 exp, 2 R_{gen} and 2 $H(\cdot)$. Any node can verify another node's self-certificate with a computational cost of 3 exp and 1 $H(\cdot)$ and only 3 exp for all subsequent verifications since the base public key has to be authenticated only once.

Self-certificate exchanges on a peer-to-peer basis (on the application and network layers) are the only communication overhead imposed on the network by the proposed scheme. A certificate exchange procedure on the application and network layer only takes 2 asynchronous rounds with 1 unicast message from each node.

5.4 Performance Evaluation Approaches

In practice there exist two main approaches to evaluate applications for wireless mobile ad hoc networks (MANETs): *simulations* and real *test beds* [123]. Both of these approaches have their advantages and disadvantages [123] [124] [125] (to be discussed shortly).

A third, less common, approach is *emulation*, which exploits the advantages of both simulations and test beds. The 'emulation' approach is unfortunately fairly novel and

places higher requirements on the memory and computational resources of the emulation/simulation server.

Theoretical analysis on communication protocols is normally used to verify the validity of simulation results. Theoretical results are obtained from deriving a model that mathematically describes the functionality of the communication protocol under investigation. Such a model should also take into consideration the characteristics of (and interaction between) the underlying communication infrastructure and the protocol under investigation. The performance of the key management scheme is directly dependent on the operation of the routing protocol, MAC protocol and node mobility model. In order to perform theoretical analysis on the proposed scheme, SelfOrgPKM, a model is needed that effectively captures the functionality of an entire mobile ad hoc network. Investigation by the author has confirmed that such a model has not been attempted in literature, the reason for which becomes apparent if the complexity of the problem is considered. This is also evident from the fact that none of the most reputable papers, discussed in sections 3.3 to 3.9, offer a comparison between a theoretical evaluation and simulation results in an attempt to validate simulation results.

In order to make an intuitive decision on which of the approaches to use for evaluating the proposed scheme, SelfOrgPKM, each feasible method will be briefly investigated.

5.4.1 Approach 1: Real Test Beds

Real test beds can provide very realistic results and to a great extent illustrate the practicality of the network based application under investigation. The quality of the results however depends heavily on how closely the test bed represents the deployment of the network in practice. Aspects to consider, in order to guarantee the validity of the results, would be (amongst others) the type of nodes that will be used in the field, the nodes' available resources (such as processing power, memory, battery life, physical size etc.), expected mobility of the users, the users network traffic profile, the number of users participating in the network, the users' behavioral patterns and the worst case networking environment.

Three disadvantages of real test beds make the approach unsuitable for evaluating network dependent applications:

- High cost of deployment.
- High labor intensity.
- Non-repeatability of results.

Take for example the evaluation of a group key management scheme designed for MANETs. Group communication in MANETs will generally occur in the form of dynamic peer groups (DPGs) and thus have group membership in the order of a hundred nodes [46] [126]. The major challenge in the field of group key management is to design protocols that are efficiently scalable and therefore can accommodate dynamic group membership, i.e. members joining and leaving at random. A real test bed for testing DPG protocols will require at the least 150 nodes as DPG sizes of 50 to 100 nodes are of interest. The considerable cost of such a test bed should be apparent. There are however other problems with test beds that make their use with respect to MANETs impractical. MANETs, due to node mobility, have rapidly changing, random network topologies. If a research institute could afford the infrastructure, each of these nodes needs to be mobile, moving at speeds of up to 20m/s (which is the maximum mobility rate used in numerous routing protocol simulations). As a consequence, besides the time and effort required to configure all these nodes, performing meaningful tests could be labor intensive and will require skillful organization. Real test beds thus pose logistic impossibilities [125]. Finally, such tests cannot be repeated since the test conditions can be affected by many factors such as channel characteristics, mobility patterns, error-prone wireless connectivity and so forth.

5.4.2 Approach 2: Simulators

Simulation offers the capability to repeat and control the network environment and to precisely specify the simulation scenario [123]. With the use of a simulation package such as ns2 [127] and GloMoSim [109], wireless network protocol designers can get a detailed

account of events that occur in the given network scenario. The most widely adopted simulator is ns2 due to its enhanced features and potential to serve as a common platform for comparing newly proposed protocols with existing protocols. By implementing the protocols within ns2, these protocols can be simulated by exploiting the lower layer models provided by the simulator.

In some cases a simulation implementation of a protocol results in an unrealistic simplification of the protocol. The results obtained from simulation may therefore significantly differ from what may be found in a "real world" application. The correctness of the results relies heavily on the implementor's understanding of the protocol and the simulator in which the protocol is implemented [128]. Secondly, the implementor may (unknowingly) exploit his/her control over the system in order to achieve a desired result. The disadvantages of simulations may inherently cast doubt over the results obtained from simulations.

5.4.3 Approach 3: Emulators

Simulations are referred to as emulations when the event scheduler needs to perform all its actions in *real time*. The ns2 simulator provides network objects and *tap* agents that allows for the injection of real network data into the simulator (emulator). It is experimentally shown that emulations can evaluate an application much more realistically than its counterparts [129] [130]. (There are also numerous other advantages to emulation with respect to real test beds and simulators that will not be considered here.) There is however one major problem with emulation that will be very difficult to overcome. Since the scheduler has to keep up with external events in real time it requires significant computational resources. Such resources can only be provided by advanced distributed and parallel processing techniques and hardware. This requirement introduces a cost factor which makes emulation in general impractical.

5.4.4 Conclusions on Protocol Evaluation Approaches

From the discussion on real test beds, simulation and emulation, it is evident that simulation is currently the only practical approach to evaluate network dependent protocols in MANETs. The ns2 simulator [127] takes preference over GloMoSim [109] due to its wide use in the MANET research community.

The proposed key management scheme, SelfOrgPKM, fortunately does not have a real time requirement and lacks the implementation complexity to make its coding internal to ns2 infeasible. The effectiveness of the proposed public key management scheme was thus investigated through simulations in the ns-2 simulator (release 2.28) [127], using the OpenSSL cryptographic library (version 0.9.7e) [131] to implement the basic modular arithmetic. The proposed scheme was coded as C++ objects as a derivative of the *application* and *agent* classes internal to ns2. The implementation supported the mathematical correctness of the cryptographic design and confirmed the low implementation complexity of the proposed scheme.

5.5 Simulation Model of SelfOrgPKM

The ns-2 implementations of the IEEE 802.11b physical layer (PHY) and medium access control (MAC) protocols were used during the simulation of SelfOrgPKM. The radio model was modified to have a nominal bit-rate of 11 Mb/sec, while supporting a 250 m transmission range.

The main objective of the simulation study was to isolate the effect of network layer certificate exchanges on the network performance under different traffic and mobility scenarios. The size of certificates was set to 460 bytes. A modification of the ns-2 constant bit-rate (CBR) traffic generator was used to simulate the connection patterns. For all simulations the CBR packet size was set to 512 bytes, with a total of 50 or 75 networking nodes. The traffic loading of the network was varied between 1 CBR packet/sec and 7 CBR packets/sec. With a total of 50 and 75 connections respectively, each node was set to have one

CBR traffic source with a single unique destination. This is also referred to as a peer-to-peer traffic pattern. In the first 90 sec of the total 1000 sec of simulation time all traffic sources are randomly started in order to force as many nodes as possible to participate in the network layer certificate exchange procedure. The network area for all simulations was set to 1000m x 1000m.

The choice of an appropriate mobility model is a problem and it is unlikely that everybody will agree with any specific choice. To be consistent with most MANET literature the random waypoint model was chosen to simulate node mobility. It can be argued that such a model does not always reflect reality. For example, Capkun *et al.* [5] present a mobility model which more accurately reflects how nodes would move in reality. This model is referred to as the *Restricted Random Waypoint* mobility model [5]. Rather than choosing its destination as a random point on the plane, a node chooses a destination point from a finite set with probability ρ and a random point with probability $1-\rho$. This model clearly fuels the rate that nodes come within "close" range since they move towards a common meeting point with a higher probability.

After extensive investigation *mobgen-ss* [132] [133] was chosen as a mobility scenario generator based on the random waypoint model. Navidi *et al.* in [132] [133] point out that the *setdest* mobility generator included in the ns-2 distribution is flawed. The initial probability distribution of *setdest* differs at a later point in time as it converges to a "steady-state distribution" [132] [133]. See [132] [133] for a detailed analysis of the *mobgen-ss* mobility model. The source code for *mobgen-ss* is available from the Toilers ad hoc networks research group (http://toilers.mines.edu).

In the simulations the mean speed was set to 5 m/sec and 20 m/sec with a 2 m/sec variance. Since a pause time greater than zero reduces the relative node speed, the pause time was set to zero. For clarity the simulation parameters are summarized in Table-5.1.

Section-4.5.5.1, explained the certificate exchange mechanism of SelfOrgPKM on the network layer. As mentioned before, the certificate exchange procedure of the proposed scheme is not bound to a specific routing protocol. The details of the routing level certificate exchange implementation is dependent on the route discovery mechanism of a

Table 5.1: Simulation parameters summary

Simulation parameter	
PHY and MAC model	IEEE 802.11b
Nominal bit-rate	11 Mb/sec
Transmission range	250 m
Routing model	Modified AODV
Certificate size	460 bytes
Simulation time	1000 sec
Node number	50 and 75
Mobility model	Random waypoint - *mobgen-ss* [132] [133]
Node mean speed	5 m/sec and 20 m/sec
Node speed variance	2 m/sec
Pause time	0 sec
Traffic	1 pkt/sec to 7 pkt/sec CBR
Number of connections	50 and 75 peer-to-peer
Traffic start time	0 - 90 sec
CBR packet size	512 bytes

suitable MANET routing protocol. The Ad Hoc On-demand Distance Vector (AODV) routing protocol [33] was chosen for the simulations. An AODV agent is conveniently included in ns-2 (release 2.28), as a wireless network layer model.

SelfOrgPKM's network layer certificate exchange procedure was integrated into the AODV protocol's route discovery mechanism. AODV allows for route discovery by forwarding a route request (RREQ) from the source node to the destination node. Each intermediate node that receives the RREQ caches the broadcast ID of the RREQ and captures the reverse route in its routing table. An intermediate node with a fresh enough route will reply to the source node with a route reply message (RREP) or forward the RREQ to its neighbors. If a node receives a RREQ and has already cached the RREQ broadcast ID or was the originator of the RREQ, it will disregard the received RREQ.

AODV's RREQ receive function was modified as follows: assume intermediate $Node_C$ broadcasts a RREQ which is received by its neighbor, $Node_D$. As usual $Node_D$ checks the broadcast ID and RREQ source address for consistency. $Node_D$ then consults its repository of received certificates to determine whether it has not already received the certificate from $Node_C$. If the query is positive, $Node_D$ has the certificate of $Node_C$ and will continue to process the RREQ as usual. If $Node_D$ has not received the RREQ before *and* did not originate the RREQ *and* does not have the certificate of the $Node_C$, two approaches are defined:

- In the first scenario, $Node_D$ takes an ***optimistic approach***: when $Node_D$ receives a RREQ from $Node_C$ it sends its own certificate to $Node_C$ and processes the RREQ as normal. If $Node_C$ receives the certificate it will reply to $Node_D$ with its own certificate.

- In the second scenario, $Node_D$ takes a more ***conservative approach***: when $Node_D$ receives a RREQ from $Node_C$ and finds that it does not have $Node_C$'s certificate, $Node_D$ will send its own certificate to $Node_C$ and disregard the AODV RREQ. If $Node_C$ receives the certificate it will reply to $Node_D$ with its own certificate. AODV is unaware that the RREQ has been received and dropped by $Node_D$ and will respond to the scenario as if $Node_D$ is unavailable.

It is stressed again that this modification of AODV is not to make the protocol secure as a whole. Of primary interest is the impact of localized peer-to-peer certificate exchange on the network performance when integrated into a practical MANET routing protocol. The two scenarios explained above are partly applicable to previous efforts to secure AODV [40]. If the first scenario is used to distribute keying material for example to SAODV, then user P_i's base public key y_i, subordinate public key y_i' and public signature parameter r_i, will have to be appended to RREQs in order for the next hop to verify the DSA or ElGamal type signatures on the RREQ messages [40]. This will result in additional routing overhead in addition to the single or double RREQ signature extensions. In the second scenario the overhead will be eliminated at the cost of losing possible routes due to dropping the RREQ messages.

5.6 Simulation Results of SelfOrgPKM

In this section the simulation results of SelfOrgPKM are presented. The aim is to make an assesment of SelfOrgPKM's impact on network performance. The following two metrics are observed:

- Constant bit-rate (CBR) packet delivery ratio (PDR) as a function of mobility, load and node density.
- CBR packet end-to-end delay as a function of mobility, load and node density.

The primary function of any communication network is to deliver data packets between end points with an acceptable success rate and tolerable delay. It is therefore important to establish if the proposed certificate exchange mechanism on the network layer degrades the performance of the network.

In Figure-5.1, the simulation results for the *optimistic* and *conservative* approaches (Section-5.5), correspond closely with the CBR reference simulation at low mobility (5m/sec mean with 2 m/sec variance). As anticipated an increase in mobility (from the mean 5m/sec to 20m/sec) degrades the overall performance of the network. Again both the simulations

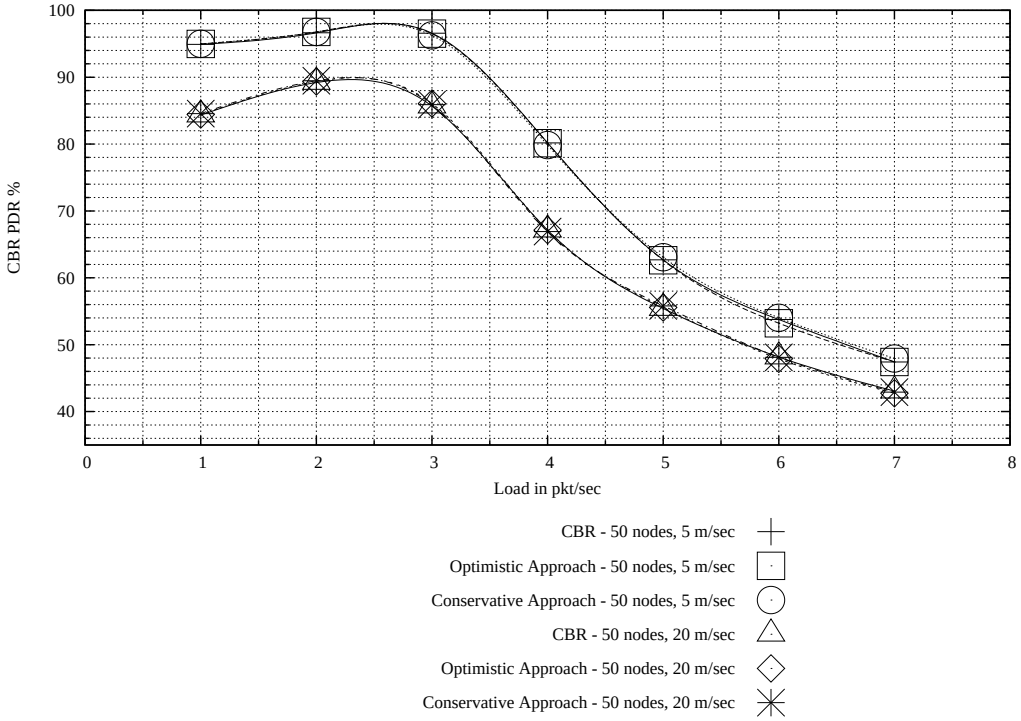

Figure 5.1: CBR packet delivery ratio (PDR) % vs load in pkt/sec for 5m/sec and 20m/sec mobility with 50 nodes

for the *optimistic* and *conservative* approaches follow the CBR reference simulation. In fact with high mobility, the correlation between all simulations becomes even closer. This supports the findings of Capkun *et al.* [5] that mobility can be an advantage in securing MANETs. The proposed scheme thus also exploits mobility to enhance the establishment of security associations.

Figure-5.2, shows the PDR vs load results for 75 nodes at a mobility of 5m/sec and 20 m/sec. With a maximum difference of 3 % PDR between the reference CBR simulation, *optimistic* approach and *conservative* approach, it is clear that the degradation in performance due to network layer certificate exchanges is insignificant[1]. Due to the localized nature of the exchange mechanism the protocol copes very well with an increase in node density. Figure-5.2, substantiates the earlier claim that mobility improves the correlation between simulations. The improved correlation as mobility increases can be explained as follows: in the given simulation setup all nodes initially do not share any security asso-

[1]See Section-5.7 for comments (8) on the scalability of the certificate exchange mechanism.

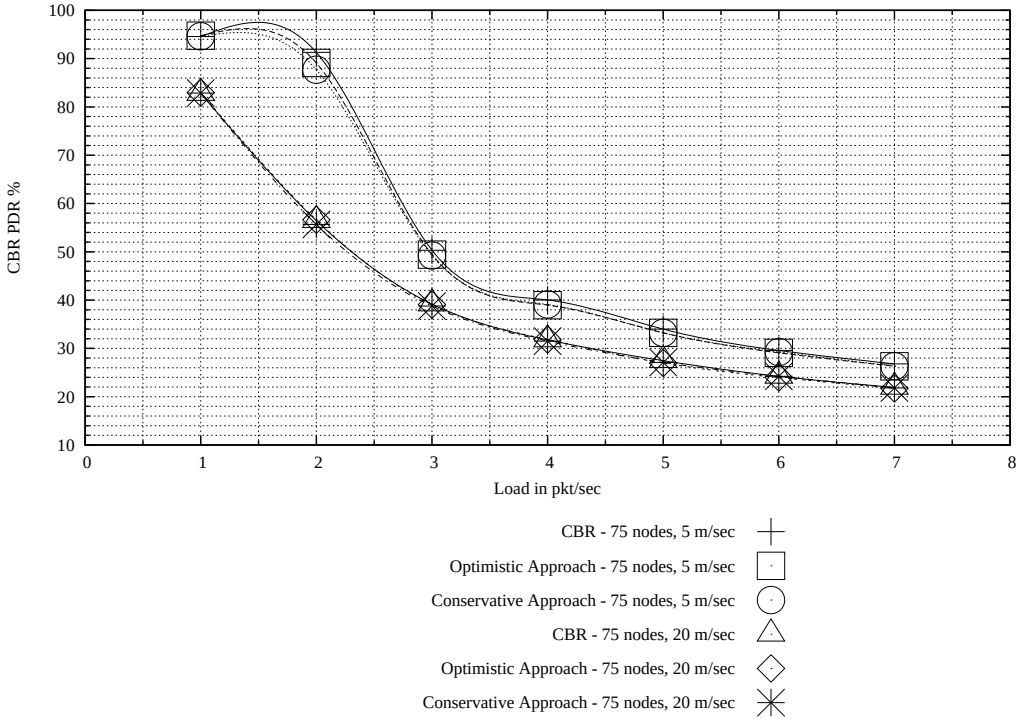

Figure 5.2: CBR packet delivery ratio (PDR) % vs load in pkt/sec for 5m/sec and 20m/sec mobility with 75 nodes

ciations. Each of the nodes starts to transmit CBR packets to one of the other nodes in the network. This triggers a flurry of certificate exchanges on the network layer. At low mobility this results in some RREQ packets being dropped on the routing layer. As shown in Figure-5.3, low mobility extends the period over which RREQs are dropped. Firstly, at higher mobility nodes are more likely to come within each others transmission range, increasing the rate at which nodes set up security associations (see Figure-5.3). This may be an indication that the proposed network layer certificate exchange mechanism would in fact perform even better with a mobility model, such as the *Restricted Random Waypoint* model [5]. Secondly, at higher mobility some nodes within each other's transmission range have already set up security associations, hence the RREQs that are dropped have less impact on AODV's ability to route packets to their destinations.

In Figure-5.2, it can be seen that as the load increases above 3 pkt/sec, with 75 nodes, the performance of the network breaks down to an unacceptable level. The breakdown in performance of the reference CBR simulation with 75 nodes supports the choice of 50

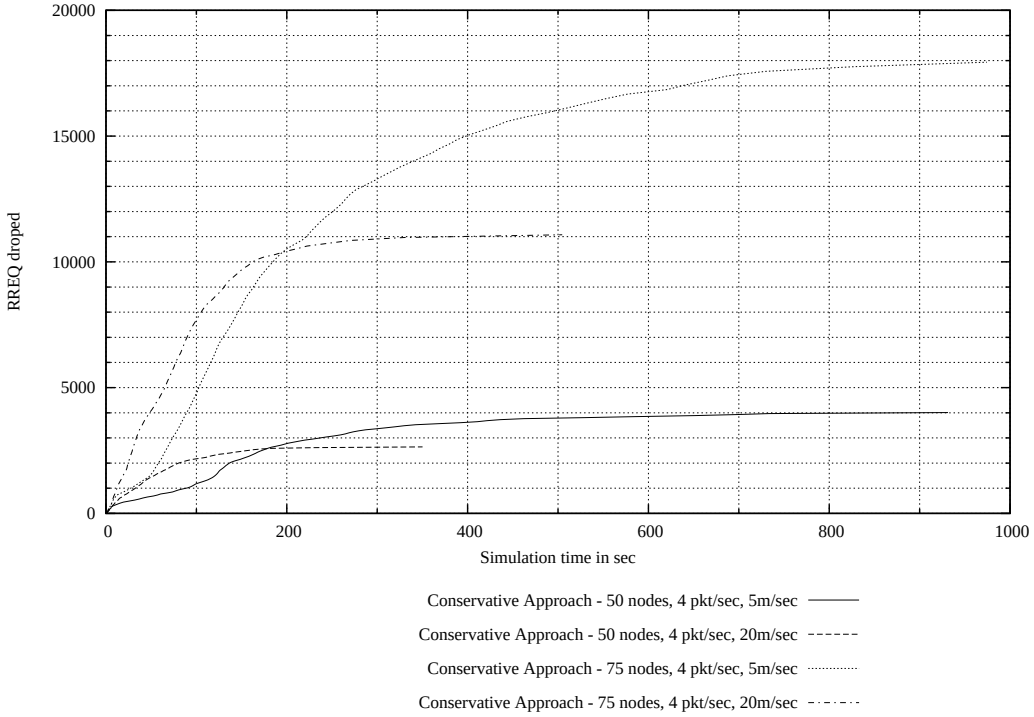

Figure 5.3: RREQ dropped vs time for a 4 pkt/sec load

nodes as the main node density. Some interesting observations will also be made from the 75 node simulation results in due course.

To place the results of PDR vs load into perspective the average end-to-end delay for CBR packets is included. Figure-5.4 shows that the exchange of certificates on the network layer does not add significant delay to the average delivery time of CBR packets.

The CBR end-to-end delay with 75 nodes, shown in Figure-5.5, gives some insight into the interaction between the AODV routing protocol and IEEE 802.11 MAC protocol. Within the given simulation model (see Section-5.5), each node has a multi-hop connection with one other node in the network. As intended, this triggers as many certificate exchanges on the network layer as possible. It will be informative to consider the CBR PDR vs load shown in Figure-5.2 together with the end-to-end delay in Figure-5.5. It is noted from Figure-5.5 that at low mobility (5 m/sec) increasing the load above 2 pkt/sec causes an exponential decrease in performance. The end-to-end delay at 5 m/sec also shows this clearly. With an average delay of 0.2 sec at a load of 2 pkt/sec per node, the end-to-end

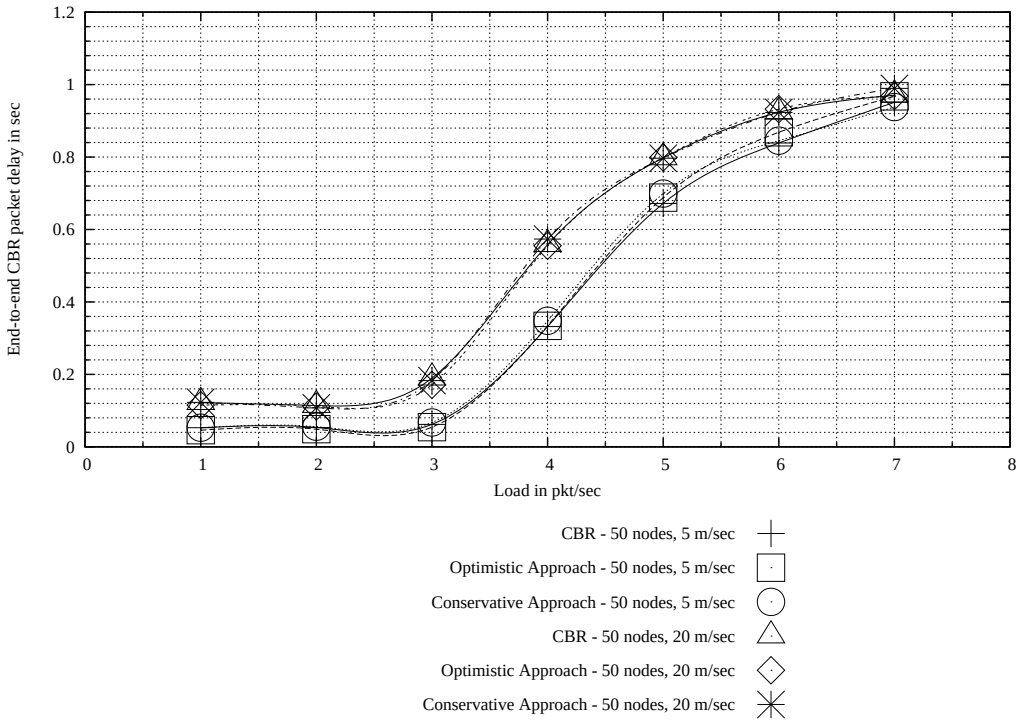

Figure 5.4: CBR packet end-to-end delay % vs load for 50 nodes

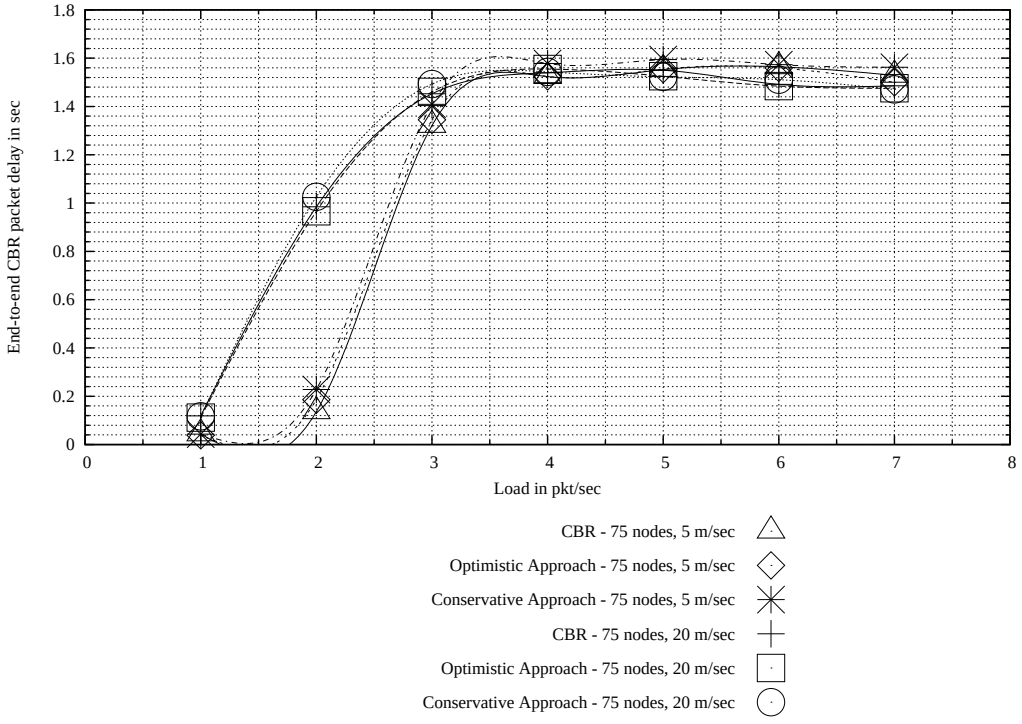

Figure 5.5: CBR packet end-to-end delay % vs load for 75 nodes

delay increases above 1.3 sec for 3 pkt/sec. Increasing the mobility to 20 m/sec worsens the situation. With a 20m/sec mobility the CBR PDR drops below 60 % at 3 pkt/sec, while the end-to-end delay increases to about 1 sec. Investigation by the author has shown that the main reason for this is congestion and packet collisions. Figure-5.6 and Figure-5.7 show the number of route requests (RREQs) sent by AODV as a function of load for 50 and 75 nodes respectively. Increasing the number of nodes from 50 to 75 causes a significant increase in the number of RREQs sent by the routing protocol. For 75 nodes the number of RREQs increases from approximately 550 000 to 1.2 million. The rapid increase in end-to-end delay and decrease in performance can be explained in terms of an avalanche effect. Increasing the number of nodes, increases the effective overall load on the network as the number of pkt/sec sent by each node is increased. This results in the AODV routing protocol initiating more route requests which increases the likelihood of congestion and collisions. The increase in congestion causes more packets to be dropped by the 802.11 MAC and the increase in collisions cause more packets to be received in error. Consequently the routes requested by AODV are not established. This causes AODV to send more RREQs which leads to more congestion and collision. As expected, this vicious cycle continues until AODV stabilizes as 802.11 MAC reaches its maximum effort. Figure-5.2, 5.5 and 5.7 show that higher mobility (20 m/sec) fuels this avalanche effect as the rate at which the performance deteriorates, increases significantly.

5.7 Design Verification

In this section it will be argued that the proposed scheme satisfies all the given design specifications in the problem statement as given in Section-4.5.1. Firstly, the generic requirements (1 to 8) will be considered:

1. SelfOrgPKM supports a *hybrid* key management scheme. Once two nodes have exchanged their self-certificates they can use their asymmetric keying material to set up a shared session key. They may want to do this for efficiency reasons.

2. In Section-5.2.2 a strong, but easy to follow, argument was provided on the explicit

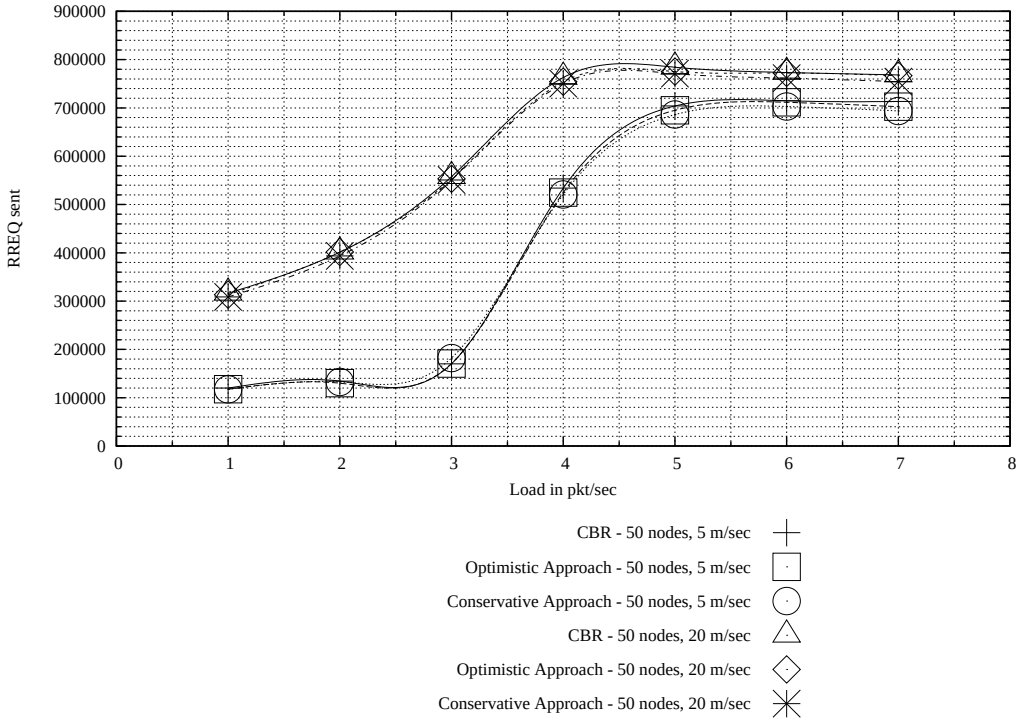

Figure 5.6: RREQ sent vs load in pkt/sec for 50 nodes

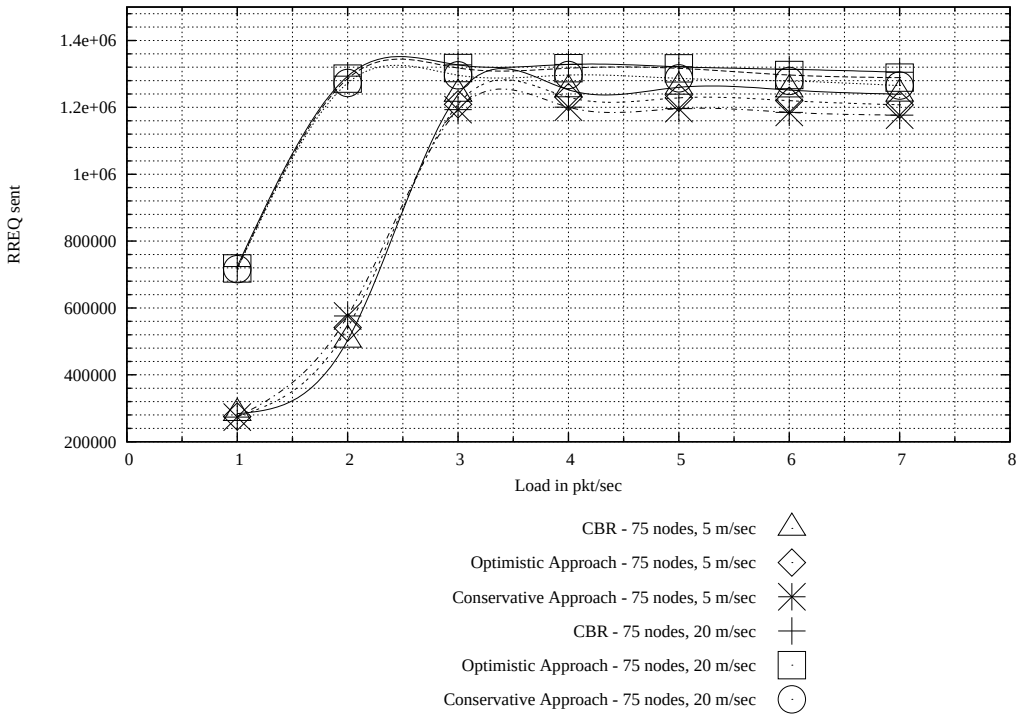

Figure 5.7: RREQ sent vs load in pkt/sec for 75 nodes

authenticity of a user's keying material.

3. The security argument in Section-5.2.2 gives evidence of key independence between a user's base private key, subordinate private key and renewed subordinate private keys.

4. The only data transmitted between users over the network is public key certificates. Section-5.2 showed that an active, insider adversary cannot forge certificates. Certificates are exchanged via the simple two-round certificate exchange mechanism explained in Section-4.5.5.1. An attacker may falsely trigger certificate exchanges between nodes by sending bogus route requests. Note that this is an issue that has to be solved by the routing protocol. Together with the property of key independence, SelfOrgPKM also provides resistance to known key attacks and hence subsumes forward and backward secrecy (see Section-5.2).

5. It is claimed that the proposed scheme will remain available as long as nodes can communicate with other nodes within their transmission range. Clearly if one-hop communication is not possible the network is of no use at all. SelfOrgPKM does not rely on the routing infrastructure for certificate dissemination. Users in an "open" wireless network are bound to come into physical contact at least once, in particular if they have data to exchange that requires high security. They can then use key establishment mechanisms as detailed in [5] [105] [106] to set up a security association. Even if they don't engage in a physical encounter they have to exchange their network addresses by some out-of-band means, which in the proposed system relates to an authentic exchange of $ID_i = H(y_i)$, where y_i is the base public key of party i.

6. SelfOrgPKM provides robustness by distributing the key management service to all network participants. Schemes that make use of an on-line authority (centralized or distributed) may fail to provide robustness. Similar to [5], SelfOrgPKM requires each node to be its own authority domain. Users therefore do not rely on any other node in the network for key management services, which minimizes the impact of hardware/software failures, network partitioning, attacks on the routing mechanisms, sporadic connectivity etc.

7. The proposed key management scheme is efficient (see Section-5.3). The communication is limited to two messages per establishment of a bi-directional security association between node pairs. From the presented simulation results it can be seen that this is however not entirely true in a realistic implementation (see Section-5.6). Due to network congestion and packet collisions some certificate exchanges on the network layer are not successful and therefore a subsequent exchange will be triggered if a route request is received from the same node at a later point in time. The efficiency analysis in Section-5.3 demonstrates that SelfOrgPKM has competitive computational requirements and outperforms most of its counterparts.

8. It is noted that the given simulation model represents the worst case scenario, since all nodes form the network simultaneously: when a network is formed, all the nodes do not normally join the communication network at the same time[2]. For example, assume an "open" MANET is formed in a shopping center; the probability that all customers will arrive at the shopping center at once, switch on their communication devices at the same time and set up a constant bit-rate data connection, is highly unlikely. In fact the author strongly agrees with Steiner *et al.* [46] that ad hoc networks will exhibit the properties of dynamic peer groups (DPGs). From the simulation results it can be seen that if the network starts off with 100 or fewer nodes (even if they join simultaneously as in the worst case simulation model) the proposed scheme will not have scalability problems even if the network grows to thousands of nodes. Unfortunately this claim is only intuitively supported as the resources to simulate such large networks are not readily available.

   In ns-2 a simulation with 75 nodes in SelfOrgPKM's simulation model produce tracefiles as large as 15 GB (with MAC and network level tracing enabled). The simulations with ten desktop computers[3] took just over 72 hours to complete. The simulation parameters in Table-5.1 combined with the scenarios explained in the simulation model, gives a total of 840 simulations. Each computer thus took six days to complete 84 simulations, which includes the time to process the tracefiles.

[2] Although SelfOrgPKM's simulation model is extreme, the effect of localized certificate exchange on the network layer could be observed with as many certificate exchanges triggered as possible.

[3] Model: HP D530, CPU: Pentium 4 2.8 GHz at 533 MHz FSB, RAM: 512MB DDR SDRAM, Hard Disk: Samsung SP-0411N 7200/133, Operating System: Linux SuSe 9.3 Pro.

Clearly increasing the node density in the given simulation model to more than 100 nodes creates logistic impossibilities.

Now that it has been shown that the proposed scheme satisfies the generic requirements for a key management scheme, the remaining self-imposed specifications (9 to 18, Section-4.5.1) is investigated:

9. As mentioned before, the proposed key management scheme only requires users to exchange their self-certificates during the on-line post-initialization phase. The use of the routing control packets to trigger the certificate exchanges avoids introducing any other message types or extending the size of the routing packets. The certificate exchange mechanism of SelfOrgPKM is completely on-demand. The proposed scheme only exchanges keying material between nodes that are required to secure the routing infrastructure in a localized context. Users exchange keying material on the application layer on a need-to-know basis if they want to communicate securely.

10. SelfOrgPKM's certificate exchange mechanism explained in Section-4.5.5.1 breaks the routing-security interdependence cycle. As pointed out in (5) above, the proposed scheme does not rely on the routing infrastructure at all.

11. The use of crypto-based identifiers [110] [45], solves the address ownership problem. To the best of the author's knowledge it is the only way to solve this problem without any form of trusted authority.

12. SelfOrgPKM's basic assumption is that each node is its own authority domain (similar to [5]). The key management scheme is not dependent on any form of on-line TTP or user collaboration. The adversary will not gain different levels of advantage by compromising different nodes. Compromising any node and any number of nodes will not effect the operation of SelfOrgPKM.

13. The simulation results discussed in Section-5.6 show that mobility does not effect the proposed scheme. In fact, higher mobility lessens the impact of the network layer certificate exchange mechanism on the network performance.

14. Although SelfOrgPKM exploits mobility, it does not rely on mobility in any way: nodes exchange certificates in a self-organized fashion and can therefore support secure routing without depending on users to have physical encounters.

15. In Section-5.6 it is shown that the proposed scheme has a negligible impact on the network performance, both in terms of packet delivery ratio and end-to-end delay. On the other hand, the routing protocol will not find any routes if the necessary security association is not established (see Section-5.5 for an explanation of the *conservative approach*). The proposed scheme therefore introduces no noticeable time delay in the setting up of of security associations.

16. In Section-5.2 the security of SelfOrgPKM is discussed. The security of the signature scheme is proved in ROM+GM and it is shown that subordinate public keys cannot be forged based on the security of the signature scheme and the intractability of the discrete logarithmic problem.

17. The proposed scheme, SelfOrgPKM has very low implementation complexity. The only cryptographic primitives required are a random number generator, the modified ElGamal signature scheme (Section-4.3) and a suitable hash function. Note that the proposed subordinate public key generation scheme (Section-4.4) is based on the modified ElGamal signature scheme. The modifications to the routing protocol is regarded as minor.

18. Lastly it is noted that the scheme is fully modularized. All the mechanisms of SelfOrgPKM (random number generator, hash function, signature scheme, subordinate public key generation/renewal scheme and certificate exchange mechanism) can be used as secure building blocks in other applications.

5.8 On the Shortcomings of SelfOrgPKM

The main deficiency of the proposed peer-to-peer key management scheme is that nodes can generate more than one identity. As mentioned in Section-2.1.3, this is however an inherent disadvantage of *fully* self-organized schemes [12].

SelfOrgPKM's certificate revocation mechanism may not appear to be extensive (see Section-4.5.5.2). Certificate revocation mechanisms for MANETs normally rely on some threshold t of nodes to agree on revoking a certificate. In the presence of adversaries with multiple identities this approach does not make much sense. Certificate revocation in *fully* self-organized MANETs is a very difficult problem. It is expected that the certificate revocation problem may be *partially* solved with an appropriate trust model and various node misbehavior detection mechanisms; in "open" networks, a malicious node can rejoin the network with a new identity once excluded.

The proposed scheme's certificate exchange mechanism focuses primarily on establishing security associations for the routing infrastructure. Self-organized application level key establishment techniques for SelfOrgPKM is not explicitly defined. SelfOrgPKM's methods need to be integrated with those defined in [5] [105] [106] to effectively set up security associations on both the network and application layer.

The certificate exchange mechanism may not provide adequate certificate dissemination for all the existing secure routing protocols. Consequently, either the certificate exchange mechanism or the routing protocol will have to be enhanced to distribute all the needed certificates. This will particularly be required in low mobility scenarios.

5.9 Conclusion

The thesis contributes a novel peer-to-peer key management scheme for fully self-organized mobile ad hoc networks, called Self-Organized Peer-to-Peer Key Management (SelfOrgPKM). The scheme has low implementation complexity and provides self-organized mechanisms for certificate dissemination and key renewal without the need for any form of off-line or on-line authority.

The fully distributed scheme is superior in communication and computational overhead with respect to its counterparts [1] [2] [65] [3]. All nodes send and receive the same number of messages and complete the same amount of computation. SelfOrgPKM therefore preserves the symmetric relationship between the nodes. Each node is its own authority

domain which provides an adversary with no convenient point of attack.

SelfOrgPKM solves the classical routing-security interdependency problem and mitigates impersonation attacks by providing a strong one-to-one binding between a user's certificate information and public key.

The thesis also introduces two generic cryptographic building blocks as the basis of SelfOrgPKM: 1) A variant on the ElGamal type signature scheme developed from the generalized ElGamal signature scheme introduced by Horster *et al.* The modified scheme is one of the most efficient ElGamal variants, outperforming most of the other variants; and 2) A subordinate key generation scheme.

The thesis introduces the novel notion of *subordinate public keys*, which allow the users of SelfOrgPKM to perform self-organized, public key revocation without changing their network identifiers/addresses. Subordinate public keys therefore eliminate the main weakness of previous efforts to solve the address ownership problem in Mobile IPv6 [110] [45]. Furthermore, the main weakness of previous efforts to break the routing-security interdependence cycle in MANETs [92] is also eliminated by using a subordinate public key mechanism. The presented ElGamal variant was proved to be secure in ROM+GM (*Theorem 1*) without making any unrealistic assumptions. It was shown how the strong security of the signature scheme supports the security of the proposed subordinate key generation scheme. The philosophy behind SelfOrgPKM's security argument is well described by Koblitz and Menezes in [119].

The only operation of SelfOrgPKM affecting the network is the pairwise exchange of certificates. The cryptographic correctness, low implementation complexity and effectiveness of SelfOrgPKM were verified though simulation using ns-2 and OpenSSL. The simulation results show that the novel, localized certificate exchange mechanism on the network layer has negligible impact on network performance. The simulation results furthermore demonstrate that network layer certificate exchanges can be triggered without extending routing protocol control packets.

Chapter 6

Conclusions and Future Direction

Mobile ad hoc networks (MANETs) have some unique characteristics that make the design of suitable security mechanisms both challenging and interesting. The security issues in MANETs will have to be resolved before these networks will find wide scale deployment. An important differentiation can be made between different types of MANETs based on the use of a trusted authority. This work focused on key management in "open" or *fully* self-organized MANETs. The lack of any form of trusted authority makes the design of a *low complexity* key management scheme for "open" MANETs a difficult task.

The first part of this thesis started with a detailed explanation of the characteristics of MANETs from a security perspective. This was followed by a brief overview of the application of MANETs, both in a military and commercial setting. These applications serve as the driving force behind numerous research initiatives. A wide range of security issues in MANETs was summarized and key management was identified as one of the main security problems.

In the second part of the thesis a detailed survey on the existing key management schemes was given. The proposals were subdivided based on their main characteristics or design approach. The survey considered schemes that make use of both off-line and on-line trusted authority or neither of the two; the discussion and comments on each subset are applicable to the design of key management scheme for "open" and "closed" MANETs.

The third part of the thesis conveyed the main contribution. A novel peer-to-peer key management scheme for fully self-organized MANETs, called Self-Organized Peer-to-Peer Key Management (SelfOrgPKM) was proposed. The scheme has low complexity and a strong security argument. The scheme is fully distributed and provides an adversary with no single point of attack. SelfOrgPKM solves the address ownership problem with crypto-based identifiers as proposed by O'Shea *et al.* SelfOrgPKM breaks the routing-security interdependence cycle with a novel certificate exchange mechanism. The key distribution scheme exploits the control packets of the routing protocol to trigger certificate exchanges on-demand. SelfOrgPKM's certificate dissemination mechanism is thus able to establish security associations in a localized context as required by the routing infrastructure. Altough SelfOrgPKM fully exploits mobility to enhance the setup of the security associations, the scheme does not rely on mobility at all. It was shown through analysis and simulations that SelfOrgPKM has negligible impact on the performance of the network. The certificate exchange mechanism introduces no noticeable time delay in providing keying material for the routing mechanism.

As building blocks for the proposed peer-to-peer key management scheme a variant on the ElGamal signature scheme was presented and a new subordinate public key generation scheme was proposed. The security of the signature scheme was proved in the Random Oracle and Generic Security Model (ROM+GM). Subordinate public keys are a novel notion which allows users to generate a secondary public key from their original or base public key. It is well known that the use of cryptographic keys degrades their security level, hence the need for a key renewal mechanism. If only the subordinate public key is used for real communication it leaves the adversary with a brute-force attack on the base private key as the only option for breaking the system.

6.1 Summary of Contributions

The following lists the contributions of the thesis:

- A comprehensive survey and analysis on the existing key management schemes for

MANETs.

- A novel peer-to-peer key management scheme for *fully* self-organized MANETs that exploits user mobility, but does not depend on mobility in any way.
- A novel certificate exchange mechanism that breaks the routing-security interdependence cycle by decoupling the key distribution scheme from the routing infrastructure. It eliminates any delay in the setup of security associations: the routing infrastructure can therefore be secured at the start of network formation.
- A novel notion called subordinate public keys which eliminates the main weakness of previous efforts to solve the address ownership problem in Mobile IPv6. Furthermore, it solves the main weakness of previous efforts to break the routing-security interdependence cycle in MANETs.
- An efficient variant of the ElGamal signature scheme, proved secure in ROM+GM.

6.2 Direction for Future Research

Future work related to key management includes the following:

- The proposed certificate exchange mechanism will be enhanced and simulated in more realistic mobility models as presented in [134]. It will also be interesting to see, for example, how large one can make the certificates before they start to degrade the network performance. The certificate exchange mechanism will be optimized to come closer to the lower bound of two messages per establishment of a bi-direction security association.
- The capabilities of an adversary with multiple identities will have to be considered before "open" MANETs will become feasible.
- System parameter agreement without any form of trusted authority or user interaction is an issue that needs investigation.

- The proposed peer-to-peer key management scheme for fully self-organized MANETs is ideally suited to securing the routing infrastructure. As mentioned before, it falls short when it comes to securing communication on the network layer. Future work includes an intergration of SelfOrgPKM with mechanisms for authenticated key agreement on the application layer.

- In order to provide a stronger security argument for the next fully self-organized key management scheme, the proposed subordinate public key generation scheme will be proved secure in the Random Oracle and Generic Security Model (ROM+GM).

- A key management scheme for self-organized MANETs has been investigated [135] [136]. It completely eliminates the need for an on-line authority and is ideally suited for "closed" or military type MANETs. It uses an off-line trusted authority to bootstrap the system and can therefore provide strong access control. An interesting feature of the scheme is that the off-line authority does not learn the private keys of users. This property may be essential in some MANET applications. Future work includes simulation of the scheme in realistic mobility models and further development of a strong security argument.

- Group key management is still an open problem in MANETs. Preliminary investigations has been completed [126] [137]. The next step is to complete simulations in order to establish the suitability of conventional wireline group key management techniques for MANETs .

- Key management in vehicular ad hoc networks is a complex open problem to be investigated in future.

Bibliography

[1] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," *IEEE Network: special issue on network security*, vol. 13, no. 6, pp. 24–30, 1999.

[2] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, "Self-securing Ad Hoc Wireless Networks," in *proc. Seventh International Symposium on Computers and Communications (ISCC'02)*, July 1-4 2002.

[3] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64, 2003.

[4] E. C. H. Ngai, M. R. Lyu, and R. T. Chin, "An Authentication Service Against Dishonest Users in Mobile Ad Hoc Networks," in *proc. IEEE Aerospace Conf.*, March, 6-13 2004.

[5] S. Capkun, J. Hubaux, and L. Buttyan, "Mobility Helps Peer-to-Peer Security," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 43–51, 2006.

[6] S. Yi and R. Kravets, "Composite Key Management for Ad Hoc Networks," in *proc. First Annual International Conf. on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04)*, August, 22 - 26 2004.

[7] L. Buttyan and J. P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592, 2003.

[8] A. C.-F. Chan, "Distributed Symmetric Key Management for Mobile Ad Hoc Networks," in *proc. 23rd Conf. of the IEEE Communications Society*, March, 7-11 2004.

[9] Z. J. Haas and S. Tabrizi, "On Some Challenges and Design Choices in Ad-Hoc Communications," in *proc. IEEE Military Communications Conf. (MILCOM'98)*, October, 18-21 1998.

[10] Z. J. Haas, J. Deng, B. Liang, P. Papadimitratos, and S. Sajama, "Wireless Ad Hoc Networks," in *Encyclopedia of Telecommunications*, J. Proakis, Ed. John Wiley, 2002.

[11] N. B. Salem, L. Buttyan, J.-P. Hubaux, and M. Jakobsson, "Node Cooperation in Hybrid Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, 2005, to appear.

[12] J. R. Douceur, "The Sybil Attack," in *proc. 1st International Workshop on Peer-to-Peer Systems (IPTPS'02)*, March, 7-8 2002.

[13] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[14] L. Buttyan, "Building Blocks for Secure Services: Authenticated Key Transport and Rational Exchange Protocols," Ph.D. dissertation, Universite Technique de Budapest, 2001.

[15] H. Taub and D. L. Schilling, *Principles of Communication Systems*, 2nd ed. New Delhi: McGraw-Hill, 1991.

[16] S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, no. 3, pp. 461 – 491, 2004.

[17] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall PTR, 2001.

[18] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[19] T. A.-M. Quazi, "Design and Implementation of an On-demand Ad-hoc Routing Algorithm for a Positional Communication System," Department of Electrical, Electronic and Computer Engineering, University of Natal," Degree Master of Science in Electronic Engineering, 2003.

[20] N. J. Dearham, "Development, Implementation and Quantification of an Ad-hoc Routing Protocol for Mobile Handheld Terminals," Department of Electrical, Electronic and Computer Engineering, University of Natal," Degree of Master of Science in Electronic Engineering, 2003.

[21] J. Jubin and J. D. Tornow, "The DARPA Packet Radio Network Protocol," *IEEE*, vol. 75, no. 1, pp. 21–32, 1987.

[22] "USA Federal Emergency Management Agency (FEMA): Information on Federally Declared Disasters," 2005. [Online]. Available: http://www.fema.gov/

[23] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. Decouto, "Carnet: A scalable ad hoc wireless network system," in *proc. 9th ACM SIGOPS European Workshop*, September, 17-20 2000.

[24] W. Franz, R. Eberhardt, and T. Luckenbach, "Fleenet - Internet on the road," in *proc. 8th World Congress on Intelligent Transport Systems*, October 2001.

[25] M. Raya and J. P. Hubaux, "The Security of Vehicular Ad Hoc Networks," in *proc. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05)*, November, 7 2005, to appear.

[26] S. Murphy and J. J. Garcia-Luna-Aceves, "An efficient routing algorithm for mobile ad hoc networks," *MONET*, vol. 1, no. 2, pp. 183–197, 1996.

[27] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-hoc Wireless Networks," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 1996, pp. 153–181.

[28] J. Sharony, "A Mobile Radio Network Architecture with Dynamically Changing Topology using Virtual Subnets," in *proc. IEEE International Conf. on Communications (ICC/SUPERCOMM'96)*, June 1996.

[29] V. D. Park and M. S. Corson, "A Highly Adaptable distributed Routing Algorithm for Mobile Wireless Networks," in *proc. Sixteenth Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM'97)*, 1997.

[30] C.-K. Toh, "Associativity-based Routing for Ad Hoc Mobile Networks," *Wireless Personal Communications: An International Journal*, vol. 4, no. 2, pp. 103–139, 1007.

[31] Z. J. Haas and M. Perlman, "The Performance of Query control Schemes for Zone Routing Protocol," in *proc. Annual Conf. of the ACM Special Interest Group on Data Communication (SIGCOMM'98)*, September 2-4 1998.

[32] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized Link State Routing Protocol," Internet Engineering Task Force MANET Work Group (IETF MANET WG)," Internet Draft, November 1998.

[33] C. E. Perkins and E. M. Belding-Royer, "Ad-hoc On-demand Distance Vector Routing," in *proc. The Second IEEE Workshop on Mobile Computing Systems and Applications (IEEE WMCSA'99)*, February 1999.

[34] T. A.-M. Quazi and S. McDonald, "A Load Aware, Location Aided, Ad-hoc On-demand Routing Algorithm," in *proc. Fifth IFIP TC6 International Conf. on Mobile and Wireless Communications Networks (MWCN'03)*, 2003.

[35] N. J. Dearham and S. McDonald, "A Location Aided Multicasting Protocol (LAMP) for Sparse Ad-Hoc Networks," in *proc. Southern African Telecommunication Networks and Applications Conf. (SATNAC'03)*, 2003.

[36] Y.-C. Hu, D. B. Johnson, and A. Perrig, "Ariadne: A Secure OnDemand Routing Protocol for Ad Hoc Networks," in *proc. Eighth ACM International Conf. on Mobile Computing and Networking (Mobicom'02)*, September 2002.

[37] ——, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," *Workshop on Mobile Computing Systems and Applications. IEEE*, 2002.

[38] B. Dahill, E. Levine, E. Royer, and C. Shields, "A Secure Routing Protocol for Ad Hoc Networks," University of Massachusetts, Technical Report UM-CS-2001-037, August 2001.

[39] P. Papadimitratos and Z. J. Haas, "Secure Routing for Mobile Ad Hoc Networks," in *proc. SCS Communication Network and Distributed System Modeling and Simulation Conf. (CNDS'02)*, January, 27-31 2002.

[40] M. Guerrero Zapata, "Secure Ad Hoc On-demand Distance Vector (SAODV) Routing," September, 15 2005, iNTERNET-DRAFT draft-guerrero-manet-saodv-04.txt.

[41] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook in Applied Cryptography.* CRC Press, 1996.

[42] B. Schneier, *Applied Cryptography*, 2nd ed. Applied Cryptography, 1996.

[43] T. Karygiannis and L. Owens, "Wireless Network Security 802.11, Bluetooth and Handheld Devices," National Institute of Standards and Technology (NIST), US Department of Commence, Technical Report NIST Special Publication 800-48, November 2002.

[44] M. Jakobsson, W. S, and Y. B, "Stealth Attacks on Ad-Hoc Wireless Networks," in *proc. Vehicular Technology Conf.*, October, 6-9 2003.

[45] G. Montenegro and C. Castelluccia, "Crypto-based Identifiers (CBIDs): Concepts and Applications," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 97–127, 2004.

[46] M. Steiner, G. Tsudik, and M. Waidner, "Key Agreement in Dynamic Peer Groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769–780, 2000.

[47] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Wormhole Detection in Wireless Ad Hoc Networks," Department of Computer Science, Rice University, Technical Report TR01-384, December 2001.

[48] J. Kong, Z. Ji, W. Wang, M. Gerla, R. Bagrodia, and B. Bhargava, "Low-cost Attacks against Packet Delivery, Localization and Time Synchronization Services in Under-Water Sensor Networks," in *proc. ACM Workshop on Wireless Security (WiSe'05)*, September, 2 2005.

[49] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols," in *proc. ACM Workshop on Wireless Security (WiSe'03)*, September 2003.

[50] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," University of Cambridge Computer Laboratory, AT and T Laboratories, Cambridge, Technical Report tr.1999.2, 1999.

[51] ——, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," in *proc. 3rd AT and T Software Symposium*, October 1999.

[52] R. Molva and P. Michiardi, "Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks," in *proc. European Wireless (EW'02)*, February, 25-28 2002.

[53] L. Buttyan and J. P. Hubaux, "Enforcing Service Availability in Mobile Ad-Hoc WANs," in *proc. IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, August 2000.

[54] ——, "Nuglets: a Virtual Currency to Stimulate Cooperation in Self-Organized Mobile Ad Hoc Networks," Swiss Federal Institute of Technology - Lausanne (EPFL-DI-ICA), Technical Report DSC/2001/001, January 2001.

[55] S. Buchegger and J.-L. Le Boudec, "Nodes Bearing Grudges: Towards Routing Security, Fairness, and Robustness in Mobile Ad-hoc Networks," in *proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, January 2002.

[56] ——, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Distributed Ad-hoc NeTworks," in *proc. Third ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, June, 9-11 2002.

[57] R. Molva and P. Michiardi, "Core: A Collaborative Reputation Mechanism To Enforce Node Cooperation In Mobile AD HOC Networks," in *proc. The 6th IFIP Communications and Multimedia Security Conf.*, September 2002.

[58] ——, "A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad hoc Networks (extended abstract)," in *proc. Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, March, 3-5 2003.

[59] H. Yang, X. Meng, and S. Lu, “”Self-Organized Network-Layer Security in Mobile Ad Hoc Networks”,” in *proc. ACM Wireless Security Workshop (WiSe'02)*, September 2002.

[60] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, “Diversity Coding for Transparent Self-healing and Fault-Tolerant Communication Networks,” *IEEE Transactions on Communications*, vol. 41, no. 11, pp. 1677–1686, 1993.

[61] G. Ateniese, M. Steiner, and G. Tsudik, “Authenticated Group Key Agreement and Friends,” in *proc. 5th ACM Confernce on Computer and Communications Security*, November, 2-5 1998.

[62] Y. Kim, A. Perrig, and G. Tsudik, “Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups,” in *proc. 7th ACM Conf. on Computer and communications security (CCS'00)*, November, 1-4 2000.

[63] ——, “Tree-based Group Key Agreement,” *ACM Transactions on Information Systems Security (TISSEC)*, vol. 7, no. 1, pp. 60 – 96, 2004.

[64] J. P. G. Sterbenz, R. Krishnan, R. R. Hain, A. W. Jackson, D. Levin, R. Ramanathan, and J. Zao, “Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions,” in *proc. ACM Workshop on Wireless Security (WiSe'02)*, September, 28 2002.

[65] S. Yi and R. Kravets, “MOCA: Mobile certificate authority for wireless ad hoc networks,” in *proc. of the 2nd Annual PKI Research Workshop (PKI 2003)*, April, 28-29 2003.

[66] A. Khalili, J. Katz, and W. A. Arbaugh, “Towards secure key distribution in Truly Ad-Hoc Networks,” in *proc. IEEE Workshop on Security and Assurance in Ad-Hoc Networks*, January, 28 2003.

[67] J.-P. Hubaux, L. Buttyan, and S. Capkun, “The Quest for Security in Mobile Ad hoc Networks,” in *proc. MobiHoc'01*, June 9-11, 2002 2001.

[68] S. Capkun, L. Buttyan, and J.-P. Hubaux, “Mobility Helps Security in Ad Hoc Networks,” in *proc. MobiHoc*, June 1-3. 2003 2003.

[69] S. Yi and R. Kravets, “Practical PKI for Ad Hoc Wireless Networks,” Department of Computer Science, University of Illinois, Technical Report UIUCDCS-R-2002-2273, UILU-ENG-2002-1717, August 2001.

[70] ——, “Key Management for Heterogeneous Ad Hoc Wireless Networks,” Department of Computer Science, University of Illinois, Technical Report UIUCDCS-R-2002-2290, UILU-ENG-2002-1734, July 2002.

[71] ——, "Key Management for Heterogeneous Ad Hoc Wireless Networks," in *proc. 10th IEEE International Conference on Network Protocols (ICNP'02)*, November, 12-15 2002.

[72] G. Xu and L. Iftode, "Locality Driven Key Management Architecture for Mobile Ad-hoc Networks," in *proc. First IEEE International Conference on Mobile and Sensor Networks (MASS'04)*, October, 24-27 2004.

[73] B. Wu, J. Wu, E. B. Fernandez, and S. Magliveras, "Secure and Efficient Key Management in Mobile Ad Hoc Networks," in *proc. First International Workshop on Systems and Network Security (SNS2005) (in conjunction with IPDPS)*, April 2005.

[74] B. Wu, J. Wu, E. B. Fernandez, M. Ilyas, and S. Magliveras, "Secure and Efficient Key Management in Mobile Ad Hoc Networks," *Journal of Network and Computer Applications*, 2005, to appear.

[75] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[76] J. Broch and D. B. Johnson, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, IETF Internet Draft," October 1999.

[77] A. Herzberg, S. Jaracki, H. Krawczyk, and M. Yung, "Proactive Secret Sharing Or: How to Cope With Perpetual Leakage," in *proc. Advances in Cryptology - CRYPTO '95*, 1995.

[78] C. Carter, S. Yi, P. Ratanchandani, and R. Kravets, "Manycast: exploring the space between anycast and multicast in ad hoc networks," in *proc. 9th Annual International Conf. on Mobile Computing and Networking (MOBICOM'03)*, September, 14-19 2003.

[79] Y. Desmedt and S. Jajodia, "Redistributing Secret Shares to New Access Structures and Its Applications," Department of Information and Software Engineering, School of Information Technology and Engineering, George Mason University, Technical Report ISSE-TR-97-01, July 1997.

[80] T. M. Wong, C. Wang, and J. M. Wing, "Verifiable Secret Redistribution for Archive System," in *proc. First International IEEE Security in Storage Workshop*, December, 11 2002.

[81] P.-A. Fouque and J. Stern, "One Round Threshold Discrete-Log Key Generation without Private Channels," in *proc. Public Key Cryptography - PKC'01*, February, 13-15 2001.

[82] R. Zhang and H. Imai, "Round Optimal Distributed Key Generation of Threshold Cryptosystem Based on Discrete Logarithm Problem," in *proc. Applied Cryptography and Network Security (ACNS'03)*, October, 16-19 2003.

[83] C.-M. Li, T. Hwang, and N.-Y. Lee, "Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders," in *proc. Advances in Cryptology - EUROCRYPT'94*, May, 9-12 1994.

[84] C.-T. Wang, C.-H. Lin, and C.-C. Chang, "Threshold signature schemes with traceable signers in group communications," *Computer Communications*, vol. 21, no. 8, pp. 771–776, 1998.

[85] W.-B. Lee and C.-C. Chang, "(t, n) Threshold Digital Signature with Traceability Property," *Journal of Information Science and Engineering*, vol. 15, no. 5, pp. 669–678, 1999.

[86] Z.-C. Li, J.-M. Zhang, J. Luo, W. Song, and Y.-Q. Dai, "Group-oriented (t, n) threshold digital signature schemes with traceable signers," in *proc. Topics in Electronic Commerce, Second International Symposium, ISEC 2001*, April, 26-28 2001.

[87] H. Pedersen, "How to convert any digital signature scheme into a group signature scheme," in *proc. 5th International Workshop on Security Protocols*, April, 7-9 1997.

[88] M. Michels and P. Horster, "On the Risk of Disruption in Several Multiparty Signature Schemes," in *proc. Advances in Cryptology - ASIACRYPT '96*, November, 3-7 1996.

[89] Y.-M. Tseng and J.-K. Jan, "Attacks on threshold signature schemes with traceable signers," *Information Processing Letters*, vol. 71, no. 1, pp. 1–4, 1999.

[90] G. Wang, X. Han, and B. Zhu, "On the Security of Two Threshold Signature Schemes with Traceable Signers," in *proc. Applied Cryptography and Network Security, First International Conf., ACNS 2003*, October, 16-19 2003.

[91] T.-S. Wu and C.-L. Hsu, "Cryptanalysis of group-oriented (t, n) threshold digital signature schemes with traceable signers," *Computer Standards and Interfaces*, vol. 26, no. 5, pp. 477–481, 2004.

[92] R. B. Bobba, L. Eschenauer, V. D. Gligor, and W. Arbaugh, "Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks," in *proc. IEEE Global Telecommunications Conference*, December 2003.

[93] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks," in *proc. Ninth International Conf. on Network Protocols (ICNP'01)*, November, 11-14 2001.

[94] D. Joshi, K. Namuduri, and R. Pendse, "Secure, Redundant, and Fully Distributed Key Management Scheme for Mobile Ad Hoc Networks: An Analysis," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 4, pp. 579–589, 2005.

[95] M. Joye and S.-M. Yen, "ID-based Secret-key cryptography," *ACM Operating Systems Review*, vol. 32, no. 4, pp. 33–39, 1998.

[96] D. Boneh and M. Franklin, "Identity-Based Encryption from Weil Pairing," in *proc. Advances in Cryptology - CRYPTO'01*, August 2001.

[97] J. C. Cha and J. H. Cheon, "An Identity-Based Signature from Gap Diffie-Hellman Groups," in *proc. Public Key Cryptography - PKI'03*, January, 6-8 2003.

[98] H. Deng, A. Mukherjee, and D. P. Agrawal, "Threshold and Identity-based Key Management and Authentication for Wireless Ad Hoc Networks," in *proc. International Conference on Information Technology: Coding and Computing (ITCC'04)*, 2004.

[99] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Secure Distributed Key Generation for Discrete-Log Based Cryptosystems," in *proc. Advances in Cryptology - EUROCRYPT'99*, May, 2-6 1999.

[100] H. Petersen and P. Horster, "Self-certified keys - Concepts and Application," in *proc. Third Conf. on Communication and Multimedia Security*, September 22-23 1997.

[101] P. Zimmermann, *The Official PGP User's Guide*. MIT Press, 1995.

[102] B. Christianson, "Why Isn't Trust Transitive," in *proc. International Workshop on Security Protocols*, April, 10-12 1996.

[103] A. Abdul-Rahman and S. Hailes, "A Distributed Trust Model," in *proc. ACM New Security Paradigms Workshop*, September, 23-26 1997.

[104] A. Josang, E. Gray, and M. Kinateder, "Analysing Topologies of Transitive Trust," in *proc. First International Workshop on Formal Aspects in Security and Trust (FAST'03)*, September 2003.

[105] M. Cagalj, S. Capkun, and J. Hubaux, "Key agreement in peer-to-peer wireless networks," *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, 2005, to appear.

[106] J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication," in *proc. IEEE Symposium on Security and Privacy*, 2005.

[107] Y. P. Chen and A. L. Liestman, "A Zonal Algorithm for Clustering Ad Hoc Networks," *International Journal of Foundations of Computer Science*, vol. 14, no. 2, pp. 305–322, 2003.

[108] T. Beth, B. Malte, and K. Birgit, "Valuation of Trust in Open Networks," in *proc. Third European Symposium on Research in Computer Security*, 1994.

[109] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a Library for Parallel Simulation for Large-scale Wireless Networks," in *proc. 12th Workshop on Parallel and Distributed Simulations (PADS '98)*, 1998.

[110] G. O'Shea and M. Roe, "Child-proof authentication for MIPv6 (CAM)," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 2, pp. 4–8, 2001.

[111] P. Horster, M. Michels, and H. Petersen, "Generalized ElGamal signatures for one message block," in *proc. 2nd Int. Workshop on IT-Security*, September, 22-23 1994.

[112] ——, "Meta-Message Recovery and Meta-Blind Signature Schemes Based on the Discrete Logarithm Problem and their Applications," in *proc. Advances in Cryptology - ASIACRYPT'94*, 1994.

[113] ——, "Hidden signature schemes based on the discrete logarithm problem and related concepts," in *proc. Communications and Multimedia Security*, 1995.

[114] M. Girault, "Self-certified public keys," in *proc. Advances in Cryptology - EUROCRYPT'91*, 1991.

[115] Y. Huang and L. Wenke, "A Cooperative Intrusion Detection System for Ad Hoc Networks," in *proc. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, October 2003.

[116] N. Ferguson and B. Schneier, *Practical Cryptography.* Wiley Publishing, 2003.

[117] C. P. Schnorr and M. Jakobsson, "Security of Discrete Log Cryptosystems in the Random Oracle and Generic Model," in *proc. The Mathematics of Public-Key Cryptography*, June, 12- 17 1999.

[118] ——, "Security of Signed ElGamal Encryption," in *proc. Advances in Cryptology - ASIACRYPT '00*, December,3-7 2000.

[119] N. Koblitz and A. J. Menezes, "Another Look at "Provable Security"," *Journal of Cryptology*, 2004, to appear.

[120] G. Montenegro and C. Castelluccia, "Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses," in *proc. Network and Distributed System Security Symposium (NDSS'02)*, February 2002.

[121] National Institute of Standards and Technology (NIST), "Secure Hash Standard," U.S. Department of Commence, Federal Information Processing Standards Publication FIPS PUB 180-1, April,17 1995.

[122] X. Wang, Y. L. Yin, and H. Yu, "Finding Collisions in the Full SHA-1," in *proc. 25th Annual International Cryptology Conference (Crypto'05)*, August, 14-18 2005.

[123] Q. Ke, D. Maltz, and D. B. Johanson, "Emulation of Multi-Hop Wireless Ad Hoc Networks," in *proc. The 7th International Workshop on Mobile Multimedia Communications (MoMuC2000)*, October, 23-26 2000.

[124] J. Flynn, H. Tewari, and D. O'Mahony, "JEmu: A Real Time Emulation System for Mobile Ad Hoc Networks," in *proc. Firsit Joint IEI/IEE Symposium on Telecommunications Systems Research*, November 2001.

[125] K. Fall, "Network Emulation in the Vint/NS Simulator," in *proc. The Fourth IEEE Symposium on Computers and Communications (ISCC'99)*, July, 6-8 1999.

[126] J. van der Merwe, D. Dawoud, and S. McDonald, "A Survey on Group Key Management for Mobile Ad Hoc Networks," *ACM Computing Surveys (CSUR)*, 2004, in submission.

[127] "The Network Simulator - ns-2," available at www.isi.edu/nsnam/ns.

[128] M. Takai, J. Martin, and R. Bagrodia, "Effects of wireless physical layer modeling in mobile ad hoc networks," in *proc. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc01)*, October, 4-5 2001.

[129] D. Mahrenholz and S. Ivanov, "Real-Time Network Emulation with ns-2," in *proc. 8-th IEEE International Symposium on Distributed Simulation and Real Time Applications*, October, 21-23 2004.

[130] ——, "Adjusting the ns-2 Emulation Mode to a Live Network," in *proc. Kommunikation in Verteilten Systemen (KVS'05)*, February, 28 2005.

[131] "OpenSSL cryptography library," available at www.openssl.org.

[132] W. Navidi, T. Camp, and N. Bauer, "Improving the Accuracy of Random Waypoint Simulations Through Steady-State Initialization," in *proc. 15th International Conference on Modeling and Simulation (MS'04)*, March 2004.

[133] W. Navidi and T. Camp, "Stationary Distributions for the Random Waypoint Mobility Model," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, pp. 99–108, 2004.

[134] J.-Y. Boundec Le and M. Vojnovic, "Perfect Simulation and Stationarity of a Class of Mobility Models," in *proc. IEEE Infocom*, March, 13-17 2005.

[135] J. van der Merwe, D. Dawoud, and S. McDonald, "A Survey on Peer-to-Peer Key Management for Military Type Mobile Ad Hoc Networks," in *proc. Military Information and Communications Symposium of South Africa*, 2005.

[136] ——, "Public Key Management for Military Type Mobile Ad Hoc Networks," in *proc. Military Information and Communications Symposium of South Africa*, 2005.

[137] ——, "Group Key Management for Military Type Mobile Ad Hoc Networks," in *proc. Military Information and Communications Symposium of South Africa*, 2005.