Security protocols for mobile ad hoc networks

Carlton R. Davis

Doctor of Philosophy

School of Computer Science

McGill University

Montreal,Quebec

August 2006

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirement for the degree of Doctor of Philosophy.

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-32171-3
Our file Notre référence
ISBN: 978-0-494-32171-3

DEDICATION

This thesis is dedicated to my wife and parents

ACKNOWLEDGEMENTS

I would like to express my deep appreciation to the following persons:

- My supervisor Prof. Claude Crépeau for the help and financial support he provided for the duration of my doctoral studies; and for the facilities he supplied to do my research.
- My co-supervisor Prof. Muthucumaru Maheswaran for the insights and guidance he provided.
- Geneviève Arboit for her assistance with the complexity and security analysis of the certificate revocation protocol; and for her help in translating the abstract into French.
- The System Staff at the School of Computer Science, in particular Philippe Ciaravola (the System Manager), Andrew Bogecho and Ron Simpson for the technical help and other favors I received from them.
- My family and friends for the support and encouragement I received.

Contributions of Authors

Most of the work presented in this thesis have been published or is being reviewed for publication in refereed journals or conferences. The contributions of the co-authors of the papers are as follows:

- My supervisors Prof. Claude Crépeau and Prof. Muthucumaru Maheswaran provided guidance and insights for the research projects, and help with the editing and proof-reading of the papers.
- Geneviève Arboit assisted me with the complexity and security analysis of certificate revocation protocol.
- All other work related to the design, simulation implementations, analyses and writing of the papers were done by me.

ABSTRACT

Mobile ad hoc networks (MANETs) are generating much interest both in academia and the telecommunication industries. The principal attractions of MANETs are related to the ease with which they can be deployed due to their infrastructure-less and decentralized nature. For example, unlike other wireless networks, MANETs do not require centralized infrastructures such as base stations, and they are arguably more robust due to their avoidance of single point of failures. Interestingly, the attributes that make MANETs attractive as a network paradigm are the same phenomena that compound the challenge of designing adequate security schemes for these innovative networks.

One of the challenging security problems is the issue of certificate revocation in MANETs where there are no on-line access to trusted authorities. In wired network environments, when certificates are to be revoked, certificate authorities (CAs) add the information regarding the certificates in question to certificate revocation lists (CRLs) and post the CRLs on accessible repositories or distribute them to relevant entities. In purely ad hoc networks, there are typically no access to centralized repositories or trusted authorities; therefore the conventional method of certificate revocation is not applicable.

Another challenging MANET security problem is the issue of secure routing in the presence of selfish or adversarial entities which selectively drop packets they agreed to forward; and in so doing these selfish or adversarial entities can disrupt the network traffic and cause various communication problems.

In this thesis, we present two security protocols we developed for addressing the above-mentioned MANET security needs. The first protocol is a decentralized certificate revocation scheme which allows the nodes within a MANET to have full control over the process of certificate revocation. The scheme is fully contained and it does not rely on any input from centralized or external entities such as trusted CAs. The second protocol is a secure MANET routing scheme we named Robust Source Routing (RSR). In addition to providing data origin authentication services and integrity checks, RSR is able to mitigate against intelligent, colluding malicious agents which selectively drop or modify packets they are required to forward.

RÉSUMÉ

Les réseaux ad hoc mobiles (ou MANETs, l'acronyme du terme anglais "mobile ad hoc networks") suscitent beaucoup d'intérêt dans le milieu académique et dans l'industrie des télécommunications. Leurs points les plus intéressants ont trait à la facilité avec laquelle ils peuvent être mis en service, vu leur nature décentralisée et sans infrastructure. Par exemple, les MANETs ne nécessitent pas d'infrastructures centralisées telles des stations de base. De plus, on peut les supposer plus robustes vu qu'ils contournent le problème de l'échec à un point unique. De façon intéressante, ce sont ces mêmes attributs des MANETs, qui les rendent attrayants comme paradigme de réseau, qui ajoutent au défi de la conception d'algorithmes de sécurité adéquats pour ceux-ci.

Un de ces défis est le problème de sécurité de la révocation de certificats dans les MANETs où il n'y a pas d'accès en ligne à des autorités de confiance. Dans le contexte des réseaux avec fils, quand des certificats doivent être révoqués, les autorités de certificats (CAs) ajoutent l'information concernant les certificats en question à des listes de révocation (CRLs) et affichent les CRLs à des endroits accessibles à cet effet ou les distribuent aux entités appropriées. Pour les réseaux purement ad hoc, il n'y a typiquement pas de réserve ou d'entité de confiance centralisés. Par conséquent, la méthode conventionnelle de révocation de certificat n'est pas applicable.

Un autre défi est le problème de la sécurité de l'acheminement de paquets en présence d'entités égoïstes ou adversaires qui, de façon sélective, ne transmettent pas les paquets qu'elles ont accepté de transmettre. Ce faisant, ces entités peuvent perturber le trafic du réseau et causer une multitude de problèmes de communication.

Dans cette thèse, nous présentons deux protocoles de sécurité que nous avons développés afin de satisfaire les besoins de sécurité des MANETs mentionnés ci-dessus. Le premier protocole est un algorithme de révocation de certificats décentralisé qui permet aux nœuds d'un MANET d'avoir le plein contrôle sur le processus de révocation de certificats. Cet algorithme est tout à fait complet en lui-même et ne dépend d'aucune information provenant d'entités externes ou centralisées, telles des CAs. Le deuxième protocole est un algorithme d'acheminement sécurisé pour les MANETs que nous avons nommé Robust Source Routing (RSR). En plus de fournir des services d'authentification d'origine ainsi que des contrôles d'intégrité, RSR peut atténuer les effets d'agents intelligents et de connivence qui, de façon sélective, ne transmettent pas ou modifient des paquets qu'ils devaient transmettre.

TABLE OF CONTENTS

CHAPTER 1
Introduction

Mobile ad hoc networks (MANETs) are autonomous collection of mobile nodes which communicate over relatively bandwidth constrained wireless links. MANETs differ from conventional wireless networks, such as cellular networks and IEEE 802.11 (infrastructure mode) networks, in that they are self-containing: the network nodes can communicate directly with each other without reliance on centralized infrastructures such as base stations. Additionally, MANETs are self-organizing and adaptive; they can therefore form and de-form on-the-fly without the need for any system administration. These unique features make MANETs very attractive for scenarios requiring rapid network deployment, such as search and rescue operations. The decentralized nature of MANETs, notably the absence of centralized entities, and hence the avoidance of single point of failures, makes these network paradigms also ideal for military and commercial applications that require high degree of robustness.

There are however some challenging security issues which need to be addressed before MANETs are ready for widespread commercial or military deployment. One of the core security issues is trust management. Trust is generally established and managed in wired and other wireless networks via centralized entities, such as certificate authorities (CAs) or key distribution centers. The absence of centralized entities in MANETs makes trust management a rather challenging problem, primarily due to the unavailability of trusted authorities to perform necessary functions such as the revocation of digital certificates. Another intriguing MANET security problem is the issue of secure routing in the presence of selfish or malicious nodes, which selectively drop packets they are required to forward; and in so doing, these selfish or malicious entities can cause various communication problems. The principal objective of this thesis is to address the above-mentioned MANET security issues.

1.1 Contributions of the thesis

The contributions of this thesis are the following:

1. A localized certificate revocation scheme which allows the nodes within a MANET to revoke certificates in a secure way, such that protection is provided against wrongful revocation of well-behaving nodes' certificates through malicious accusations. We evaluate the scheme via:
   (a) A security analysis
   (b) A communication complexity analysis and
   (c) Simulation assessments
2. A robust, secure routing protocol for adversarial MANET environments that are likely to contain intelligent malicious or selfish entities which selectively drop or modify packets they agreed to forward. We assess the protocol via:
   (a) Analyses and
   (b) Simulation evaluations
3. A review and analysis of the cryptographic tools that are currently used in MANET security schemes.

4. A review of the state of art of MANET security.
5. A comprehensive review of the state of art of MANET routing security.

1.2 Organization of the thesis

The thesis consists of eight chapters. Chapter one overviews some of the distinguishing features of MANETs, highlights the security issues the thesis addresses and summarizes the contributions of the thesis. Chapter two contains a review of the cryptographic tools that are currently utilized in MANET security schemes; this chapter also contains overviews of existing MANET security schemes. Chapter three highlights the routing approaches in MANETs and provides a comprehensive review of the state of art of MANET routing security. Chapters four and five formalize the research problems the thesis addresses, analyze the existing security proposals and justify the needs for the security protocols we developed. Chapters six and seven present overviews, detailed design and analyses of our protocols. The final chapter contains concluding remarks.

CHAPTER 2
Review of the state of the art of MANET security

The security requirements of MANETs are similar to that of other networks. They can be briefly summarized as follows:

- *Access control:* The need to restrict access of network resources to legitimate authorized entities.
- *Authentication:* Guarantee of the authenticity of the network peers and traffic source; that is, provides some assurance that a given network node is actually who it claims to be, and that any given network traffic actually originated from the source it purports to originate from.
- *Integrity*: Accounts for whether a given data has been modified in transit from its source to the destination.
- *Confidentiality:* Provide assurance that data in its un-encrypted form will be restricted to legitimate entities which have the authority to access the data.
- *Availability:* Network resources should be available to authorized entities without excessive delays.

Security solutions proposed for addressing access control, authentication, integrity and confidentiality services for MANETs utilize the following technologies: symmetric-key cryptography, digital certificates, and threshold public-key cryptography. In this chapter, we present a survey of proposed security solutions which employ these technologies.

2.1 Symmetric-key based solutions

We categorized the existing symmetric-key based security schemes for Wireless LAN (local area networks) into two categories: (1) IEEE 802.11 related standards and (2) other symmetric-key based proposals.

2.1.1 IEEE 802.11 related standards

Wired Equivalent Privacy (WEP) protocol is perhaps the most widely known symmetric-key based wireless network security scheme. WEP is the security mechanism incorporated in IEEE 802.11 WLAN [54]. WEP utilizes a secret key k, shared by all the communicating peers to secure data traffic. When a node needs to send a message M to a network peer n_i, it first compute a CRC-32 checksum on M, denoted as $c(M)$. $c(M)$ is then concatenated with M to give the plaintext $P = \langle M, c(M) \rangle$. Next, a 24-bit initialization vector (IV) v is selected, and the RC4 stream cipher along with the secret key k and v are used to generate a keystream, denoted as RC4(v, k). Finally, the plaintext P is exclusive-or with $RC4(v, k)$ to produce the ciphertext $C = P \oplus RC4(v, k)$, which is transmitted along with v, to n_i. To decrypt the ciphertext C, the reverse operation is performed; that is, the keystream $RC4(v, k)$ is generated and the decrypted plaintext P' is obtained by the following operation: $P' = (C \oplus RC4(v, k))$. P' is equal to P, since $P' = C \oplus RC4(v, k) = (P \oplus RC4(v, k)) \oplus RC4(v, k) = P$. The recipient can then verify the checksum by splitting P' in the form $\langle M', c' \rangle$ and check whether $c(M')$ matches the received checksum c'.

WEP has been proven to be insecure; consequently, the IEEE 802.11i [57] standard was developed as a replacement. **IEEE 802.11i** defines the Temporal

Key Integrity Protocol (TKIP) and Counter Mode CBC-MAC Protocol (CCMP). A brief description of each is outlined below.

TKIP: was designed as a short term replacement for WEP. The overall encryption process for TKIP is similar to that of WEP, but TKIP has the following enhancements.

- *Employs a Message Integrity Code (MIC)*: Instead of utilizing a CRC checksum—which offers very little protection against adversarial modification—for integrity checks, TKIP employs a light weight MIC[1] called Michael [39]. Michael is a key hashing function which employs a 64-bit key to produce a 64-bit message digest for input data of any given length.
- *Longer encryption key*: TKIP, like WEP, uses the RC4 encryption algorithm. However, as opposed to WEP which accepts encryption keys of length as short as 40 bits, TKIP requires a 128-bit key.
- *Frequent key change*: TKIP stipulates that every packet must be encrypted with a new encryption key which has not been used previously. The per-packet encryption keys are generated by a key mixing function which takes as input a base key, the node's MAC address and the packet sequence number, and outputs a 128-bit packet encryption key. The base key can be a pre-shared secret or an authentication key.

[1] MIC is commonly referred to as Message Authentication Codes (MAC); however, since IEEE 802 designated the acronym MAC for Media Access Control, MIC is used here instead.

- *Longer IV*: TKIP requires a 48-bit IV: twice the length of that of a WEP IV.
- *Optional key management provision*: TKIP has two modes of authentication: pre-shared secrets or IEEE 802.1X [55] based authentication. 802.1X is an IEEE standard for port-based authentication, access control and key management. TKIP as the framework for utilizing 802.1X for authentication and key management.

CCMP: is IEEE long term security solution for wireless LAN. CCMP provides stronger security than TKIP. It has the following features:

- *Entails a strong cryptographic algorithm*: CCMP utilizes AES [87] in Counter mode with CBC-MAC (CCM) mode [118]. CCM mode involves two techniques: Counter mode (CTR mode) [33] for confidentiality protection and Cipher Block Chaining Message Authentication Code (CBC-MAC) [86] for integrity protection. Consequently, the same 128-bit cryptographic key is used for confidentiality and integrity protection.
- *No need for per-packet keys*: The use of AES eliminated the need for frequent key changes.
- *48-bit IV*: Like TKIP, CCMP employs a 48-bit IV to provide protection against replay.
- *Optional key management provision*: As is the case with TKIP, CCMP also has the framework to use 802.1X for key management.

2.1.2 Other symmetric-key based proposals

Stajano and Anderson [110] proposed the idea of using imprinting to set up secure transient association between peers in an ad hoc network. Imprinting is a biological phenomenon; the example used in [110] is a new-born, for example a duckling emerging from it's egg, recognizes as its mother the first moving object it sees that makes a sound, irrespective of what it looks like. The comparison is made with a device—whose egg is a shrink-wrapped box enclosing it as it comes out of the factory—will recognize as its owner the first entity that sends it a secret key. The authors further recommended that the medium of physical electrical contact be used to transfer the secret keys during the imprinting phase.

Balfanz *et al* [7] proposed an extension of Stajano and Anderson "Duckling" model [110] that allows the exchange of secret cryptographic information via special location-limited side channels. The secret information can then be used to authenticate key exchange protocols utilize to set up session keys or other keying material on the wireless peers. The authors assert that the information transfer over the location-limited channel can be used instead of digital certificates for authentication; therefore negating the need for global public key infrastructure (PKI) and CAs.

2.2 Digital certificate based proposals

Symmetric-key cryptography has much lower computational overhead compared to other cryptographic technologies. The big drawback of symmetric-key cryptography is that key management can be quite tedious since the secret keys need to be exchanged over secure channels. Diffie and Hellman in their seminal paper entitled “New Directions in Cryptography,” [32] presented the concept of public-key cryptography which offers an effective solution to the key exchange problem associated with symmetric-key cryptography. Public-key cryptography, also commonly referred to as asymmetric-key cryptography, involves key pairs where the private key is kept secret and the associated public key is made public. The private keys are used for decrypting or signing data whereas the public keys are utilized for encrypting or verifying signatures. As an example, if Alice has a message she wishes to send to Bob and she wants Bob to be able to determine whether or not the message has been modified, and be able to verify that the message indeed came from her; Alice can sign the message with her private key and attach her public-key to the message before sending it to Bob. When Bob receives the message, he can verify the signature using the attached public key. If the signature verification fails, this is strong evidence that the message was modified while it was transmitted from Alice to Bob. However Bob will not be able to affirm that the message indeed came from Alice because the exchange is susceptible to impersonation attack[2] since an adversary (Eve) can intercept the message, changes it, then signs it, attaches her public key and sends the modified message to Bob. When Bob receives the message, if there were no transmission error, the signature verification will succeed; Bob would therefore be fooled in believing that the message came from Alice, when in fact it was sent by EVE.

[2] Also referred to as man in the middle attack.

As a solution to the possibility of impersonation attacks when public keys are exchanged, Diffie and Hellman introduced the idea of utilizing a central authority—they called a Public File—for storing public keys. If we employ this concept in the example above, when Alice generates her key pair, she sends her public key to a Public File; when Bob needs to verify the signature of a message from Alice or encrypt a message to send to Alice, Bob can query the Public File to ascertain Alice's public key. The Public File Diffie and Hellman proposed, needs to be universally available and is likely to be plagued with performance issue. In an effort to prevent the performance problem associated with the Public File, Loren Kohnfelder invented a construct he called *certificate* [70]. Kohnfelder defined a certificate as a digitally signed data record containing a name and a public key. Certificates by virtue of the fact that they are digitally signed, they can be held by non-trusted parties and pass around from person to person. This resolved the performance issue associated with the Public File, since this construct negates the need for all certificates to be stored in a central directory.

There are four main types of digital certificates in use today: X.509, PGP, SPKI/SDSI and KeyNote certificates. We give a brief description of each type below.

X.509 Certificates

The X.509 standard [59] was developed by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T). X.509 was originally designed to support X.500 directory [58] which include the specification for Distinguish Name (DN). A DN is a hierarchical name which can be assigned by some central global naming authority; it was intended as a means for specifying a person or thing uniquely. The X.509 standard delineated digital certificates to bind DN of a person or a device to its public key. X.509 certificates utilize a hierarchical trust model. In this model, there is a root Certificate Authority (CA) which issues certificates to delegated CAs and the CAs in turn issue certificates to end users or other CAs. A certificate is verified if it has not expired or revoked and there is a valid certificate chain traceable back to the root CA. For example, if a CA CA_i, delegated by a root CA CA_R, issued a certificate to Bob, to verify Bob's certificate, one needs to first ascertain that CA_i's certificate is valid, then verify that Bob's certificate has not expired or revoked and it was indeed issued by CA_i. This requires access to the public keys of CA_i and CA_R, and up-to-date certificate revocation information issued by CA_i and CA_R.

PGP Certificates

Pretty Good Privacy is an email and file encryption application created by Phil Zimmermann [133]. A PGP certificate differs from an X.509 certificate in two ways:

1. A PGP certificate binds a keyholder common name and email address to a public key; whereas an X.509 certificate binds a DN (distinguish name) to a public key.
2. A PGP certificate uses the web-of-trust model. In this trust model, there is no hierarchical structure. Certificates are issued and managed by end users; each end user is a CA in her own right. End users can also vouch for other users. For example, if Alice trusts Bob and Bob trusts Eve and issued a certificate to her, Bob can vouch for Eve and get Alice to sign Eve's certificate. Hence a certificate can have one or several signatures.

To verify a certificate, one need to ascertain that the certificate has not been revoked and find a certificate chain—associated with the given certificate—traceable to a user that she trusts. So for example, if Alice wishes to verify John's certificate, if Alice does not directly trust any of the signatories of John's certificate, his certificate nonetheless will be accepted if it has not been revoked and any of the signatories of John's certificate issued certificate to a user that Alice trusts, or the user issued certificate to another user who Alice trusts, and so on. In other words, John's certificate will be verified if it has not expired or revoked and there is a traceable certificate chain from his certificate to the certificate of a user who Alice trusts.

SPKI/SDSI Certificates

SPKI/SDSI is a trust management scheme comprising of two frameworks: "Simple Public Key Infrastructure" (SPKI) [36] and "A Simple Distributed Security Infrastructure" (SDSI) [102]. The SPKI/SDSI standard was developed by a IETF (Internet Engineering Task Force) work group as an alternative to X.509 and PGP certificates. The primary purpose of SPKI/SDSI certificates is authorization rather than authentication. SPKI/SDSI certificates bind either names or explicit authorizations to keys or other objects. As is the case for PGP certificates, SPKI/SDSI certificates can be issued by anyone; but unlike PGP certificates, deterministic certificate chains are used to verify the validity of SPKI/un-encryptedSDSI certificates.

KeyNote Certificates

KeyNote [10] a trust management system which evolved from a framework called PolicyMaker [11]. KeyNote and SPKI/SDSI certificates are similar in that they bind authorization or names to keys, their emphasis is on authorization rather than authentication, and issuing of certificates is not restricted to hierarchical CAs. They differ in their mode of operation mainly in the fact that KeyNote certificates contain decision code that gives explicit "yes" or "no" answer regarding the validity of the certificates; whereas the validation mechanism for SPKI/SDSI certificates requires certificate chains as input.

2.2.1 Schemes with no preference for certificate type

The majority of the proposed MANET security schemes involving digital certificates work with any of the above certificates types. These proposals can be grouped in the following categories:

(a) Certificate revocation is not addressed

(b) Certificate revocation mechanism require access to on-line certificate authorities (CAs)

(c) Certificate revocation mechanism do no require access to on-line CAs.

We present a brief overview of a selection of these schemes below.

Proposals which do not address certificate revocation

Venkatraman and Agrawal [114, 115] proposed an authentication scheme for ad hoc networks. This scheme relies on a cluster based architecture, where the network is partitioned into clusters: each cluster has an elected cluster head which maintains cluster membership information and acts as the certificate authority (CA) for its cluster. With regard to key distribution, the scheme stipulates that when a node joins a network, it is given a public and private system key pair. All the nodes in the network share this key pair. In Addition to the system key pair, each node gets a cluster key, generated by the cluster head and shared by all the nodes within a cluster. Cluster heads have all the above mentioned keys plus a unique public/private key pair which is used for exchanging session keys for communicating peers. The scheme relies on the assumption that all the nodes of a network mutually trust each other. This scheme does not address the issue of key revocation.

Eronen *et al* [37] proposed a trust model for ad hoc Jini services. Jini [5] is a Sun Microsystem technology which seeks to simplify the connection and sharing of network devices and services. When Jini is installed on a network device, it announces itself, provides information about the capabilities of the device and make itself available for connections from other Jini enabled devices. The trust model Eronen *et al* proposed for securing Jini services uses digital certificates for authenticating Jini enabled devices and for authorizing access to Jini services. This scheme does not address certificate revocation.

Messerges *et al* [83] presented a security design for general multihop ad hoc networks based on IEEE 802.15.4 [56] low-rate wireless personal area network standard. The design employs both symmetric and asymmetric-key cryptography. Elliptic curve asymmetric-key cryptography is utilized to establish symmetric keys on communicating peers. The symmetric keys are in turn used with AES [87] encryption algorithm for providing confidentiality and integrity services. This security design proposal does not address the issue of certificate revocation.

Keoh *et al* [67, 68] proposed a policy-based security framework to facilitate the establishment, evolution and management of MANETs. In this framework, a MANET is considered as a community, where the community *doctrine* is a specification which clearly defines the role of the participants in the community and the rules or policy governing their behavior. The authors defined a doctrine as an information model comprising the tuples $\langle R, P, S, TK, Sig \rangle$, where R denotes the role type of the participating user in the community; P defines a set of policies regulating the behavior of the participants assigned to the roles; S defines the constraint of the community; TK denotes the public-keys of the credential issuer; and Sig is the signature of the credential issuer. The security framework uses certificates as the basis of a participant gaining access to a community. This framework does not address certificate revocation.

Proposals which require access to trusted third party

Morogan and Muftic [84] proposed a certificate management scheme for ad hoc networks. The scheme assumes that periodic access to on-line CAs is available such that information about revoked certificates can be ascertained from CAs. When on-line access to CAs is not available, the scheme stipulates that a node security policy determines whether certificates can be accepted.

Verma *et al* [116] presented a progressive authentication scheme. This scheme utilizes digital certificates as the basis for establishing partial trust, which can be elevated or decremented based on the behavior profile of the nodes involved. The scheme requires periodic access to on-line certificate authorities (CAs) to obtain certificate revocation information. The authors proposed two models to address certificate revocation for intervals when access to on-line CAs is not available. The first model is the *Probabilistic Model.* In this model, a newly issued certificate has a trust value of 1 associated with it. A distrust value p $(0 < p < 1)$ is subtracted from the trust value each time that revocation information needs to be ascertained and on-line access to CAs is unavailable. If the trust value of the certificate falls below a certain threshold, a node can refuse to accept the certificate. When access to the CAs resumes, the trust value of certificates that have not been revoked will return to 1. The second model, called the *Weight Model* can be used in conjunction with the Probabilistic Model. Both models assume that a node has multiple certificates which are disclosed during the progressive authentication procedure. In the Weight Model, each certificate participating in a trust negotiation is assigned a weight. Certificates are accepted provided their weight is greater than a threshold value ω, where $\omega = \frac{\sum P_i * W_i}{\sum W_i}$, such that P_i and W_i are the trust value and weight of certificate i, respectively.

Proposal which does not require access to trusted third party

Candolin and Kari [19] presented a model for a security architecture for ad hoc networks operating in hostile environments. The security architecture consists of a trust management framework which utilizes digital certificates as the basis of trust. The scheme allows the certificate of a node to be revoked if a single node declares that the node in question is compromised.

2.2.2 Schemes which require PGP certificates

As outlined in Section 2.2, PGP certificates employs the web-of-trust trust mode. This model is very decentralized in nature; consequently, some researchers considers PGP certificates to be ideal for application in MANET security protocols. The MANET security proposals which involve PGP certificates are similar in principle, in the sense that they embraced the methodology of certificates being issued and managed by end-users rather than centralized authorities. Example of these proposal includes [51, 52, 12, 20].

2.3 Threshold cryptography based solutions

The idea of (k, n) threshold scheme was introduced by Shamir in [108]. A (k, n) threshold scheme allows a secret, for example a certificate authority (CA) signing key Y, to be split into n shares such that for a certain threshold $k < n$, any k components can combine and generate a valid signature; whereas $k - 1$ or fewer shares is unable to do so. Shamir threshold scheme is based on polynomial interpolation. It can be summarized as follows: A dealer with a secret $Y \in \mathbb{Z}_p$, where p is a prime and $p > n$, can divide Y into n shares by choosing a random polynomial $f(x)$ with coefficients in $\mathbb{Z}_p$, of degree at most $k-1$, satisfying the condition $f(0) \bmod p = Y$. The dealer then assigns a private share $y_i = f(x_i) \bmod p$ to each participant P_i. Any k shares can be used to find the coefficients of the one and only polynomial $f(x)$ of degree $k-1$ which passes through the k points $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$, with distinct x_i's, using the Lagrange formula:

$$f(x) = \sum_{i=1}^{k} (\prod_{j \neq i} \frac{(x - x_j)}{(x_i - x_j)}) y_i$$

The secret Y can then be found by evaluating $f(0)$ since $Y = f(0) \mod p$.

Verifiable secret sharing

In Shamir's scheme, a misbehaving dealer can deal inconsistent shares. This concern can be addressed by verifiable secret sharing (VSS), introduced by Chor, Goldwasser, Micali and Awerbuch in 1985 [25]. VSS allows the recipients of shares to verify whether or not the shares are consistent. In 1987, Feldman presented a practical verifiable secret sharing scheme [38]. Feldman (k,n) threshold VSS scheme involves the following steps:

1. The dealer chooses a random polynomial $f(x)$ with coefficients $f_0, f_1, ..., f_{k-1}$ in $\mathbb{Z}_q$, of degree at most $k-1$, such that the secret to be shared is $K = f(0) \bmod q$.
2. The dealer computes the public commitment check g^{f_i} for $i = 0, 1, ..., k-1$, broadcasts them to all the participants and sends the value $y_i = f(i) \bmod q$

secretly to participant P_i. Note that p and q are large primes such that q divides $p-1$, and $g \in \mathbb{Z}_p$ of order q.

3. Each P_i verifies whether its share is consistent by checking the following equation:
$$g^{y_i} \overset{?}{=} \prod_{j=0}^{k-1} (g^{f_j})^{i^j} \bmod p$$
4. If the equation holds, P_i broadcast a message indicating that its share is correct; otherwise, it rejects the share and inform the others via a broadcast.

Proactive secret sharing

Security of (k, n) threshold secret sharing scheme is based on the assumption that throughout the entire life of the secret, an adversary will be restricted to compromising less that k shares. This assumption may not be realistic for active, persistent, mobile adversaries. Herzberg *et al* [44] proposed a proactive secret sharing scheme allowing shares to be renewed, such that knowledge of the old shares is useless for attacking the secret after the shares are renewed. With this scheme, in order to discover a secret, an attacker needs to compromise at least k out of n shares, within a configurable time period t (hours, days or weeks), rather than having the entire life of the secret to carry out the exploit.

The basic form of Herzberg *et al* scheme uses Shamir secret threshold sharing primitive. This however only provides protection against passive adversaries which are unable to disrupt the predetermined protocol. For security against active adversaries controlling one or more of the communicating peers, verifiable secret sharing must be utilized. Herzberg verifiable share renewal protocol involves the following steps:

1. Each participant P_i (for $i = 1, .., k$) in the threshold secret sharing scheme, chooses a polynomial $f_i(x)$ of at most $k - 1$ degree with random coefficients $f_{i1}, ..., f_{ik-1}$ in $\mathbb{Z}_q$, such that $f_i(0) = 0$.
2. P_i computes the public commitment values $p_{ij} = g^{f_{ij}} \bmod p$, for $j = 0, ..., k - 1$, signs and broadcasts them to all participants; then computes $s_{ij} = f_i(j) \bmod q$, signs it and sends it secretly to participant P_j, for $j = 1, ..., k$ such that $i \neq j$.
3. P_i verifies the correctness of the shares s_{ji} it received from the participants, for $j = 1, ..., k$ such that $j \neq i$, by checking the equation:
$$g^{s_{ji}} \overset{?}{=} \prod_{j=0}^{k-1} (p_{ij})^{i^j} \bmod p$$
4. If the equation in step 3 above holds for all $k - 1$ s_{ji} values, P_i broadcasts a signed acceptance message announcing that all the checks were successful.
5. If all the participants broadcast acceptance messages, P_i proceeds to update its share $y_i^{(t-1)}$ to obtain its new share $y_i^{(t)}$ by doing the following computation:
$$y_i^{(t)} = y_i^{(t-1)} + (\sum_{j=1}^{k-1} s_{ji}) \bmod q$$
and erases all the variables except the current share $y_i^{(t)}$
6. If the equation in step 3 above doesn't hold for any of the s_{ji} it received, P_i broadcasts a signed accusation against the participant(s) associated with the irregularities.

Identity-based cryptography

Researchers ([69, 77]) have utilized identity-based cryptography in combination with threshold cryptography in the design of MANET security protocols. It is fitting therefore to give an overview of identity-based cryptography before we review the proposals which employ threshold cryptography.

Shamir [108] introduced the idea of identity-based cryptosystem in 1984. In this cryptographic scheme, there is no need to generate a public/private key pair and publish the public key; instead, a public key can be an arbitrary identity string such as an email address, IP address or any other identity info. So for example, if Alice wishes to send Bob an encrypted email, she does not need to have Bob's public key certificate; she can encrypt the email message using Bob's email address as the encryption key. When Bob receives Alice's email he contacts a trusted third party known as a Private Key Generator (PKG), provides proof of his identity and receives a private key which allows him to decrypt messages encrypted using his email address as the encryption key.

An identity-based encryption scheme consists of four randomized algorithm:
Setup: generates system parameters and a master-key. The system parameters include a description of the finite message space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$. These parameters can be publicly known but it is necessary that only the private key generator (PKG) knows the master-key.
Extract: uses the master-key to generate the private key corresponding to a public key identity string.
Encrypt: encrypts messages using the public key identity string.

Decrypt: decrypts messages using the corresponding private key.

Shamir presented an actual identity-based cryptosystem in his 1984 paper [108]; however, Boneh and Franklin [14] demonstrated the first provable secure identity-based encryption scheme. We give an overview of Boneh-Franklin identity-based encryption scheme below.

The security of Boneh-Franklin identity-based encryption scheme is based on the *Bilinear Diffie-Hellman* (BDH) Assumption. The BDH problem is as follows. Let $\mathbb{G}_1$, $\mathbb{G}_2$ be two groups of prime order q. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ be a bilinear map (a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ is bilinear if $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$) and let P be a generator of $\mathbb{G}_1$. The BDH problem in $\langle \mathbb{G}_1 \mathbb{G}_2, \hat{e} \rangle$ is: given $\langle P, aP, bP, cP \rangle$ for some $a, b, c \in \mathbb{Z}_q^*$, compute $W = \hat{e}(P, P)^{abc} \in \mathbb{G}_2$.

The four algorithms which constitute Boneh-Franklin identity-based encryption scheme is as follows:

Setup: Let IG be a BDH parameter generator satisfying the BDH assumption and let k be a security parameter given to the setup algorithm. Setup proceeds in the following steps.

Step 1: Run IG on input k to generate two prime order groups $\mathbb{G}_1, \mathbb{G}_2$ of order q and a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$. Choose an arbitrary $P \in \mathbb{G}_1$

Step 2: Pick a random $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$

Step 3: Choose four cryptographic hash functions $H_1 : \{0, 1\}^* \longrightarrow \mathbb{G}_1^*$, where $\mathbb{G}_1^*$ denotes the set $\mathbb{G}_1^* = \mathbb{G}_1 \setminus \{O\}$ where O is the identity element in the group $\mathbb{G}_1$. $H_2 : \mathbb{G}_2 \longrightarrow \{0, 1\}^n$ for some n, $H_3 : \{0, 1\}^n \times \{0, 1\}^n \longrightarrow \mathbb{Z}_q^*$ and

$H_4 : \{0,1\}^n \to \{0,1\}^n$. The message space is $\mathcal{M} = \{0,1\}^n$. The ciphertext space is $\mathcal{C} = \mathbb{G}_1^* \times \{0,1\}^n$. The system parameters are

params $= \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_{pub}, H_1, H_2 \rangle$. The master-key is $s \in \mathbb{Z}_q^*$.

Extract: Given a string ID $\in \{0,1\}^*$, the algorithm does:

(1) computes $Q_{ID} = H_1(\text{ID}) \in \mathbb{G}_1^*$ and

(2) sets the private key d_{ID} to be sQ_{ID} where s is the master-key.

Encrypt: To encrypt $M \in \{0,1\}^n$ under the public key ID, do the following:

(1) compute $Q_{ID} = H_1(\text{ID}) \in \mathbb{G}_1^*$,

(2) choose a random $\sigma \in \{0,1\}^n$,

(3) set $r = H_3(\sigma, M)$, and

(4) set the ciphertext to be $C = \langle rP, \sigma \oplus H_2(g_{ID}^r), M \oplus H_4(\sigma) \rangle$,

where $g_{ID} = \hat{e}(Q_{ID}, P_{pub}) \in \mathbb{G}_2$

Decrypt: Let $C = \langle U, V, W \rangle$ be a ciphertext encrypted using a public key ID. If $U \notin \mathbb{G}_1^*$ reject the ciphertext. To decrypt C using the private key $d_{ID} \in \mathbb{G}_1^*$ do:

(1) compute $V \oplus H_2(\hat{e}(d_{ID}, U)) = \sigma$,

(2) compute $W \oplus H_4(\sigma) = M$,

(3) set $r = H_3(\sigma, M)$. Test that $U = rP$. If not reject the ciphertext.

(4) Output M as the decryption of C.

2.3.1 Proposed security schemes involving threshold cryptography

A notable application of threshold secret sharing is threshold digital signatures. In a threshold digital signature scheme, a signing key is divided into n shares. Any k share holders can collaborate to compute a valid signature by combining the partial signatures each of the k participants generated. The partial signatures computed by applying the shares s_i to a message M are public values; and therefore they can be transmitted over insecure channels. Robust threshold digital signature schemes have been proposed for both RSA and discrete log based digital signature algorithms [109, 41].

The idea of utilizing threshold cryptography to distribute trust in ad hoc networks was proposed by Zhou and Haas in [131]. The authors articulated that the challenges associated with key management services in ad hoc networks can be resolved by distributing CA's duties amongst the network nodes. For example, a CA signing key can be partitioned into n shares and distributed to n nodes. Any k of the n nodes could then collaborate to sign and issue valid digital certificates; whereas a coalition of $k - 1$ or less nodes would not be unable to do so. The issue of certificate revocation was not addressed in this proposal.

Kong *et al* [73] presented a self-initialization protocol for handling dynamic node membership, such that new nodes can be initialized by k neighbors, and in so doing, the new nodes are given shares of the CA signing key, so that they can participate in the process of issuing certificates. The protocol stipulates that in the bootstrapping phase of the network, each node n_i is given valid certificate and the associated private key, along with a secret share S_i of the CA signing key. Any given S_i can be used in collaborate with $k - 1$ other S_i values to generate valid certificates. The protocol self-initialization scheme allows a node to compute a partial secret share of its S_i value and transmit it to an uninitialized node, such that the uninitialized node j can compute its secret share S_j, using Lagrange interpolation (Section 2.3), provided it obtained k partial secret shares. The protocol is built upon Shamir's threshold scheme and it does not involve verifiable secret sharing. It however, employs Herzberg *et al* proactive secret sharing scheme for protection against persistent adversaries. With regards to the issue of certificate revocation, the authors specified that when a certificate is deemed to be compromised, a signed counter-certificate is flooded over the network to denote that the given certificate has been revoked.

Luo *et al* [80, 72] presented an extensions of Kong *et al* work [73]. The proposal involves a framework for parallel share updates, and an improved certificate revocation mechanism. The parallel share updates builds on Herzberg *et al* scheme [44]. However, unlike the latter, which requires each node to collect inputs from all the other nodes before its new share can be computed, the authors stipulated that firstly a coalition of k nodes update their shares using Herzberg *et al* methodology; then the coalition of k nodes can update the shares of the remaining nodes utilizing the self-initialization scheme employs in [73]. This therefore allows parallelization, and consequently a more efficient share update process. The certificate revocation mechanism can be briefly described as follows: Each node n_i maintains a certificate revocation list (CRL). An entry in the CRL consists of an accused node's ID and a list of the node's accusers. If a node's accuser list contains less than k legitimate accusers, the node is marked as "suspect". Otherwise, the node in question is considered by node n_i to be misbehaving or compromised, and is marked as "convicted". A node can also designate a neighboring node n_j as been "convicted" if by its observation n_i deems n_j to be misbehaving or compromised. In such case, n_i broadcasts an accusation against n_j. When a node n_i receives an accusation against any given node, n_i first checks if the accuser is a convicted node in its CRL; if it is, the accusation is discarded; otherwise, it updates its CRL with the relevant information. When a node is delineated as being convicted, it is removed from all accuser list. A convicted node is re-classified as being suspected if its number of accusers falls below k.

Zhou *et al* [132] developed a fault-tolerant secure on-line certification authority called COCA. COCA utilizes multiple servers and its security is based on the assumption that at most t of $3t + 1$ servers can be compromised. Every client request sent to the COCA system is processed by a quorum of $2t + 1$ servers and every certificate is replicated on multiple servers. COCA is an implementation of the threshold secret sharing concepts Zhou and Haas proposed [131]. COCA also employs proactive secret sharing to provide protection against persistent adversary.

Khalili *et al* [69] presented a key distribution scheme which utilizes identity-based and threshold cryptography. In this scheme, at the time of network formation, a master key is shared amongst n participating nodes. The shares of the master key is then used to generate a master public key PK for an identity-based cryptosystem. This setup allows the network nodes (current and future) to use their identities as their public keys. A node can acquire the private key corresponding to its identity by obtaining and combining shares from any k of the n nodes which has a share of the master key. This proposal is similar in spirit to that of Zhou and Haas [131] except that it uses threshold cryptography to distribute the private key generation service (PKG) rather than the certificate authority (CA) services. This approach eliminates the necessity to distribute the public keys of the MANET nodes since their known identities (IP address, email address, etc) is used as their public keys.

Wang *et al* [117] proposed a self-managed heterogeneous certification scheme. This scheme employs threshold cryptography to distribute CA services amongst multiple nodes. It differs from the previously mentioned schemes in the following ways: (1) It allows multiple heterogeneous distributed CA systems to coexists in a MANET; whereas, the schemes reviewed above facilitate a single distributed CA system per MANET. (2) It requires each node in a network to have a physically unforgeable identification (example: an ID recorded in a smart card) as proof that it received an original certificate. This provides a measure of protection against Sybil attack [34] where nodes forge their identity and acquire multiple certificates. (3) It outlines a process which allows communicating peers to find a CA system they both trust.

Lehane *et al* [77] presented an implementation of shared threshold RSA key generation protocol which allows nodes to collaborate and distributively generate RSA key pairs. The protocol utilizes two separate techniques for generating a public key and the shares for corresponding private key: it employs Boneh-Franklin [13] techniques for generating a shared public key, and Catalano *et al* [21] protocol for computing inverses over a shared secret modulus to derived the shares for the corresponding private key. The protocol was implemented and ran on a wireless local area network consisting of two 500 Mhz laptops and a 200 Mhz Compaq

IPAQ pocket PC. The implementation results indicated that a 512 bit RSA key, an average took 2.5 minutes to generate.

Yi and Kravets [123] presented a composite key management framework which combines the ideas of a distributed (virtual) CA (using threshold cryptography) and a non-hierarchical certificate trust model such as PGP web-of-trust model [133]. The distributed CA is similar to that of Zhou and Haas [131] except that virtual CA role is restricted to a small number of nodes. The remaining nodes (the non-virtual CA nodes) individually issue and manage certificates in a similar manner as Capkun *et al* [20] scheme. The virtual CA—by virtue of the fact that it is more trusted than the non-virtual CA nodes—is used to increase the confidence level of a web-of-trust certificate chain.

Xu and Iftode [121] proposed a locality driven key management architecture. The architecture envisions a MANET as a group of interacting subnetworks. Each subnetwork establishes a distributed CA using threshold cryptography. A distributed CA issues certificates to the nodes in its subnetwork and provides public key authentication services for its community. The distributed CAs of the different subnetworks build trust relations (using threshold cryptography) with each other. These trust relations are utilized to authenticate "foreign" certificates issued by other CAs.

CHAPTER 3
Review of the state of the art of MANET routing security

In an ad hoc network, all the nodes may not be within the transmission range of each other; hence, nodes are often required to forward network traffic on behalf of other nodes. Consider for example the scenario in Fig 3–1, if node S sends data to node D, which is three hops away, the data traffic will get to its destination only of A and B forward it.

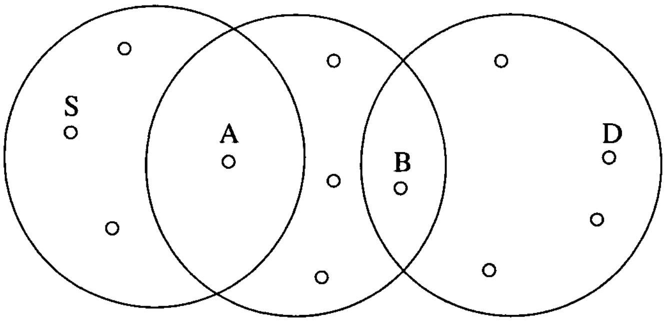

Figure 3–1: Multihop scenario

The process of forwarding network traffic from source to destination is termed routing.

3.1 Overview of routing approaches in MANETs

There are two general categories of MANET routing protocols: topology-based and position-based routing protocols. We present a brief overview of each group below. Before proceeding, it is fitting to list some desirable qualitative properties of MANET routing protocols. This list is adopted from an Internet Engineering Task Force (IETF) MANET Working Group memo [26].

- Loop-free: It is desirable that routing protocols prevent packets from circling around in a network for arbitrary time periods.
- Demand-based operation: In order to utilize network energy and bandwidth more efficiently, it is desirable that MANET routing algorithms adapt to the network traffic pattern on a demand or need basis rather than maintaining routing between all nodes at all time.
- Proactive operation: This is the flip-side of demand-based operation. In cases where the additional latency—which demand-based operations incur—may be unacceptable, if there are adequate bandwidth and energy resources, proactive operations may be desirable in these situations.
- "Sleep" period operation: It may be necessary—for reasons such as the need for energy conservation—for nodes to stop transmitting or receiving signals for arbitrary time periods. Routing protocols should be able to accommodate sleep periods without adverse consequences.
- Security: It is desirable that routing protocols provide security mechanisms to prohibit disruption or modification of the protocol operations.

3.1.1 Position-based routing protocols

Position-based routing protocols employ nodes' geographical position to make routing decisions. In order to utilize a position-based routing protocol, a node must be able to ascertain the geographical position of itself and that of all the nodes it wishes to communicate with. This information is typically obtained via Global Positioning System (GPS) and location services.

The emphasis of this thesis is on topology-based rather than position-based routing; however, we give a brief overview below of basic position-based routing algorithms.

Greedy

The Greedy routing algorithm was developed by G. Finn [40]. In the greedy forwarding approach, a node selects for the next hop, the node that is closest to the destination of the packet. In Figure 3–2, if S has data traffic to send to D which is outside of its transmission range, greedy forwarding dictates that S sends the traffic through B since B is the node within S transmission range which is closest to the destination node D.

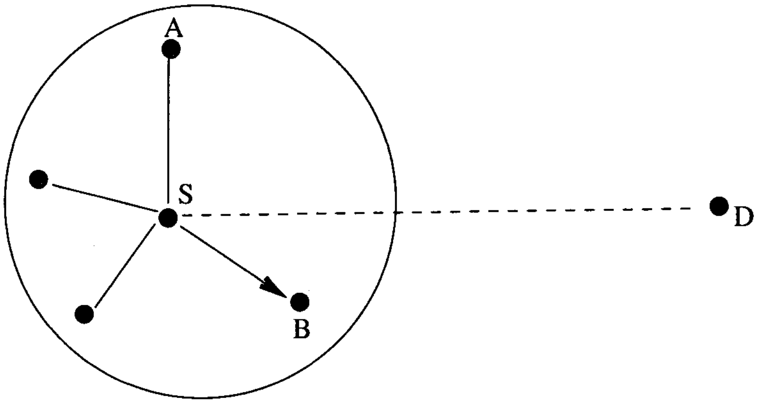

Figure 3–2: Greedy forwarding

Compass

The Compass routing algorithm was developed by Kranakis *et al* [74]. In the Compass routing scheme, a node S which has data traffic to send to a destination node D, forwards the traffic to its neighbor N which has the smallest angle $\angle NSD$, where N is a neighboring node to the forwarding node S and D is the destination. So for example in Figure 3–3, S forwards the traffic for D to A since the angle $\angle ASD$ is smaller than any other angle $\angle NSD$ where N is a node within S transmission range. Notably, Stojmenovic and Lin [111] showed that the Compass algorithm is not loop-free.

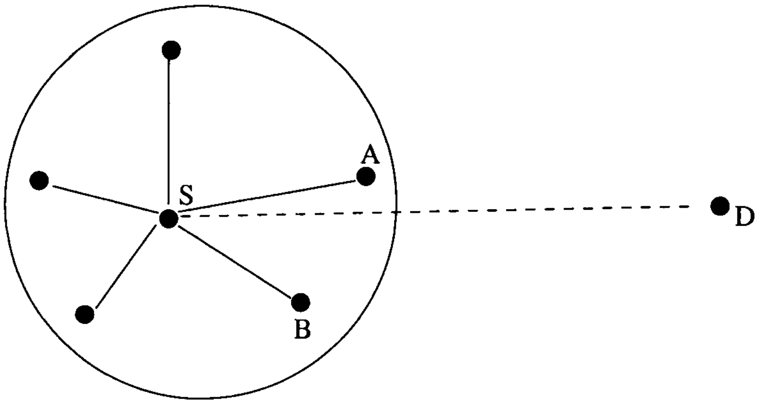

Figure 3–3: Compass forwarding

Randomized compass

The Randomized Compass routing algorithm [15] is a variation of the Compass algorithm which avoids loops with random decisions. Consider a line between a node S and a destination node D. The Random Compass forwarding approach chooses the next hop for a packet by randomly selecting between the nodes N_i and N_j which has the smallest angle $\angle N_iSD$ and $\angle N_jSD$ between the imaginary line $\overline{NS}$ (between a node N and the forwarding node S) and $\overline{SD}$, above and below the imaginary line $\overline{SD}$, respectively. So for example in Figure 3–3, node S would randomly select node A or B for forwarding packets to D since $\angle ASD$ is the smallest angle (between a line connecting S and a node that is within S transmission range, and the line $\overline{SD}$) above the line $\overline{SD}$ and $\angle BSD$ is the smallest angle below the line $\overline{SD}$.

Most Forwarded within Radius (MFR)

Takagi and Kleinroc proposed MFR [112]. Consider an imaginary line $\overline{SD}$ between a node S and a destination node D; in MFR forwarding, S forwards data traffic for D to a node A which maximizes the progress along the imaginary line $\overline{SD}$. A is therefore the node which minimizes the dot product $\overline{DA} \cdot \overline{DS}$. So in Figure 3–4, S forwards packets for D to A since A is the node within S transmission range which provides the most progress along the line $\overline{SD}$.

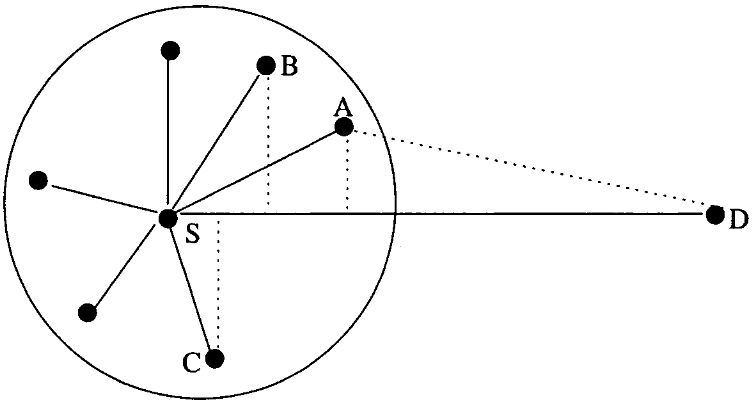

Figure 3–4: MFR forwarding

3.1.2 Topology-based routing protocols

There are two major categories of topology-based MANET routing protocols: On-demand and proactive protocols. In the section, we briefly describe some of the more prominent existing MANET topology-based routing protocols. We commence with proactive protocols.

Proactive protocols

Proactive protocols are also referred to as periodic protocols. The most prominent proactive MANET routing protocol is Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) [95]. DSDV utilizes the classical Distributed Bellman-Ford Distance-Vector algorithm [8, 62]. In distance-vector algorithms, each node i, for each destination x, maintains a set of distances $\{d_{ij}^{x}\}$, where j ranges over the neighbors of i. The distances are typically interpreted as the number of hops from i to x via the given neighbor j. Node i designates a neighbor k as the next hop for a packet if d_{ik}^{x} equals $min_j\{d_{ij}^{x}\}$. The succession of the next-hop chosen in this manner leads to x along the shortest path. In order to keep the estimated distances up-to-date, each node monitors the costs of its out-going links and periodically broadcasts to each of its neighbors, its current estimate of the shortest path to all other nodes in the network. It is well known that the Distance-Vector routing algorithm outlined above is not loop-free [23]. The main cause of routing loop formation is the fact that nodes choose their next-hops in a distributed fashion based on information which may be stale and therefore incorrect. DSDV avoids the Distance-Vector looping problem by tagging each routing distance info with a sequence number so nodes can quickly distinguish new routes from stale ones and consequently avoiding the formation of routing loops.

In DSDV routing, each MANET node maintains a routing table which is use for making routing decisions. A DSDV routing table lists all available destinations and the number of hops to each. Each routing table entry is tagged with a sequence number originated from the destination node. DSDV protocol requires each network node to advertise (via broadcasting or multicasting) to each of its current neighbors, its own routing table. Additionally, each node is required to transmit updates immediately when significantly new information is available. The routing information data a node broadcasts contains a new sequence number and and the following info for each new route:

- The destination's address;
- The number of hops from the source to the destination; and
- The sequence number of the information received regarding the destination, as originally stamped by the destination.

The MANET nodes use the advertised routing tables info and the transmitted updates to update their routing tables; which is utilized by the Distance-Vector algorithm outlined above to determine the next-hop for a packet.

On-demand protocols

On-demand protocols are also referred to as reactive protocols. Unlike proactive protocols which seeks to maintain routes to all destination in a MANET, on-demand protocols establish routes on a per need basis. There are a larger collection of existing on-demand protocols compare to proactive protocols. We present brief description of some of the more widely known on-demand protocols below.

DSR

Dynamic Source Routing (DSR) was developed by Johnson and Maltz [61]. Its basic operation is as follows: when a node S has a packet to send to a destination

D, S checks its routing cache for an entry containing a path to D. If there is no such entry, S broadcasts a routing request (RREQ) packet containing the initiator address, a unique request id , the destination address and a route record field. The latter is used to accumulate the sequence of hops the RREQ packet takes as it propagates through the network. When a node n_i receives a RREQ packet, if it has previously seen a RREQ packet with the same initiator address and request id, it discards it; otherwise, if n_i is not the destination and its routing cache does not contain a valid path to D, it records the initiator address and request id, appends its address to the route record and forwards the packet. If n_i is the destination, it returns a copy of the route record in a route reply (RREP) packet to the initiator. If n_i is not the destination but it knows of a path to D, it sends a copy of the path in a RREP packet back to the source of the RREQ packet. On receiving the RREP packet, S records the ascertained route to D in its routing cache, writes the route in the source route field of the packet header and sends the packet to the node which is the next hop in the path to D. The intermediate nodes on the path to D will likewise use the route recorded in the source route field of the packet header to determine the address of the next hop they should forward the packet to, until the packet eventually reaches the intended destination.

SSA

Signal Stability based Adaptive Routing (SSA) was developed by Dube *et al* [35]. SSA utilizes signal strength and stability of individual MANET nodes as routing selection criteria. The rational being (in the authors' view) that links which exhibit the strongest signal for the maximum amount of time leads to longer-lived routes and less route maintenance. In SSA routing, a source S sends out a route discovery request when it has data to send to a destination D that is not in its routing table. S broadcasts the route request to all its neighbors. Each neighboring node propagates the route request if it received it over a strong channel and the request has not been propagated previously. A channel is characterized as strong or weak based on the average signal strength at which the packets are exchanged between the nodes at either end of the channel. The route search packet continues to traverse the network until it reaches the destination, and it stores the address of each intermediate node it traversed. The first route search packet which arrives at the destination D is selected and a route reply packet is constructed and returned to S using the selected route. Each intermediate node in the selected route, on receiving the route reply packet, includes the new next-hop, destination pair in its routing table.

ABR

C-H Toh developed the Associativity-Based Routing (ABR) [113]. ABR utilizes the observation that a mobile node's association with its neighbor changes as it migrates and its transiting period can be identified by the associativity "ticks". Associativity ticks are updated by the mobile node's data-link protocol which periodically transmits beacons identifying itself and updates its associativity ticks in accordance with the mobile nodes in its neighborhood. A mobile node exhibits high associativity ticks (high association stability) with its neighbors when it is in a state of low mobility. Conversely, a state of high mobility is associated with low associativity ticks. In ABR routing, a node S which desires a route to a destination D broadcasts a broadcast query (BQ) message which propagates through the MANET in search of a node which has a route to the given destination. When an intermediate node n_i receives a BQ message it has not previously seen, n_i appends its address, associativity ticks with its neighbors, its relaying load, link propagation delay and its hop count to the appropriate fields of the BQ, and broadcasts the BQ to its neighbors. The next succeeding intermediate node will then erase its upstream node's neighbors' associativity ticks entries and retain only those concerning itself and its upstream nodes. When the destination node D receives the BQ packets, it selects a route based on the following selection criteria: routes consisting of nodes with higher associativity ticks has higher preference even over routes with smaller number of hops. For routes with equal number of associativity ticks, the route with the smaller hop count is selected. If the routes have the equal number of associativity ticks and hop counts, one of the route is randomly selected. The selected route is used to construct a REPLY packet and returned to the source S via the selected route. The intermediate nodes on the route from D to S will consequently mark their routes to D as valid and subsequently inactivate all other possible routes to D.

TORA

Temporally-Ordered Routing Algorithm (TORA) was developed by Park and Corson [92]. It is a highly adaptive multipath, loop-free, distributed routing algorithm which was designed for highly dynamic MANET environments. A key design concept of TORA is the localization of routing control messages to a small set of nodes near the topological change. In TORA routing, each node, at any given point in time has an associated ordered quintuple consisting of the following elements: (1) a logical time of link failure (2) the unique ID of the node which defined the new reference level (3) a single bit which is used to divide each of the unique reference level into two unique sub-levels (4) a propagation ordering parameter and (5) the unique ID of the node. Conceptually, the quintuple represents the height of a node defined by a reference level and a delta with respect to the reference level. The reference level is represented by the first three values of the quintuple while the last two values represent the delta. Each node i (other than the destination) maintains its height H_i which is initially set to NULL, $H_i = (-,-,-,-,i)$. The height of the destination is always ZERO, $H_{DID} = (0,0,0,0,Did)$, where DID represents the destination ID. In addition to its own height, each node maintains an height array with an entry $HN_{i,j}$ for each of its neighbor j. Each node i also maintains a link-state array for each of its links. The state of a link is determined by its height H_i and HN_i and is directed from higher node to lower node.

When a node requires a route to a destination D it sends out a QRY packet. When a node i receives a QRY packet it has not previously seen, it reacts as follows: (a) i rebroadcasts the QRY packet if it has no downstream links; (b) if the receiving node has at least one downstream link and its height is NULL, it sets its height to the minimum height of it non-NULL neighbors and broadcasts a UPD packet (which consists of a destination ID and the height of the node i which is broadcasting the packet); (c) if the receiving node has at least one downstream link and its height is non-NULL, it first compares the time the last UPD packet was broadcast to the time the link over which the QRY packet arrived was active. If the link became active prior to the broadcasting of the UPD packet, i discards the QRY; otherwise, i broadcasts a UPD packet. When a node i receives a UPD packet it has not previously seen from a neighbor j, i updates the entry $HN_{i,j}$ in its height array with the height contained in the UPD packet, then do the following: if its height is NULL, i sets its height to the minimum height of its non-NULL neighbor, updates all the entries in its link-state array then rebroadcasts the UPD packet which contains its new height. The process (broadcasting of QRY and UPD packets) continues until a directed acyclic graph (DAG) rooted at the destination (i.e. the destination is the only node with no downstream links) is formed. The DAG represents a route from the source S to the destination D.

AODV

Ad-hoc On-demand Distance Vector Routing was designed by Perkins and Royer [96]. Its operation can be summarized as follows: Each node using AODV maintains a route table entry for each destination of interest. A route table entry contains the destination D, next hop, number of hops to D, sequence number of the destination and the expiration time for the route table entry. When a node S has a packet to send to a destination D, S checks its routing table for an entry containing D as the destination with a sequence number equal to or greater than the last known destination sequence number of D. If there is no such entry, S broadcasts a route request (RREQ) packet, containing the source address, the source sequence number, broadcast id, destination sequence number and hop count. The source sequence number and the broadcast id are separate counters that are maintained by each node. A node increments its broadcast id counter each time it constructs a new RREQ packet; whereas the node's sequence number counter is incremented less frequently. The destination sequence number is the last known sequence number of the destination. When a node n_i receives a RREQ packet it has not previously seen, it sets up a reverse path to the source by recording the address of its neighbor from which it received the first copy of the RREQ. If n_i is not the destination and its routing table does not contain an entry for D, it increments the hop count and rebroadcasts the RREQ packet to its neighbors. If n_i however is the destination or if its routing table contains an entry with D as its destination with a destination sequence number that is equal to or greater than the destination sequence number in the RREQ packet, it constructs a route reply (RREP) packet and unicasts it to the neighboring node it received the RREQ from. An RREP packet contains the source address, destination address, destination sequence number, hop count and lifetime. When an intermediate node receives a RREP packet, it updates its routing table with the information the RREP contains, then unicasts it to the neighbor it received the first copy of the associated RREQ packet. The process continues until the RREP packet gets to S. S can now forward its packet to the next hop on the path to D.

3.2 Secure MANET routing proposals

The protocols we reviewed in Section 3.1.2 were designed for non-adversarial environments, where the node within a network are non-malicious, unselfish and well-behaving. The reality however is that in any network, there are likely to be malicious or selfish, miss-behaving nodes which have intentions of disrupting the routing protocol. Security mechanisms are therefore necessary to mitigate against these eventualities. This section reviews some of the routing security schemes which have been proposed to address the security shortcomings of these protocols. For the purpose of the review we categorized the existing secure MANET routing proposals into the following categories: basic routing security schemes, trust-based routing schemes, incentive-based schemes and schemes which employs detection and isolation mechanisms. Below, we briefly describe a selection of schemes which fall in these categories.

3.2.1 Basic routing security schemes

The routing schemes which fall in this category provide authentication services which guard against modification and replaying of routing control messages, but they do not attempt to provide solutions for issues such as the dropping of packets by selfish or malicious nodes. We commence the review with one of the earlier proposals.

Binkley and Trost presented an authenticated link-level ad hoc routing protocol [9] which was integrated into the Portland State University implementation of Mobile-IP[1] [94]. The protocol uses ICMP router discovery message [30] to discover mobile-IP nodes. It extended the ICMP router discovery packet format to include the MAC (Media Access Control) and IP address of the sender, and authentication info that can be used to verify the broadcast beacon. The protocol requires nodes

[1] Mobile-IP is a network-layer protocol which enables a mobile node to retain a fixed IP address even when it changes its point of connectivity to the Internet.

to have shared secret keys for generating message authentication codes which are used to authenticate the routing control messages.

Venkatraman and Agrawal introduced an inter-router authentication scheme [115] for securing AODV [96] routing protocol against external attacks (such as impersonation attacks, replaying of routing control messages and certain denial of service attacks). The scheme is based on the assumption that the nodes in the network mutually trust each other and it employs public key cryptography for providing the security services. The integrity of routing requests are ensured by the originating node hashing the messages and signing the resulted message digest. Recipients of a route request can check its authenticity and integrity by computing the hash of a the message using the agreed upon hash function, compare the computed hash with that attached to the message and verifying the signature. "Strong authentication" is provided for adjacent pair of nodes which transmit route replies. The strong authentication procedure is as follows: A node n_i sends a pre-reply plus a random challenge (challenge1) to a neighbor it wishes to send a reply. The neighbor n_j which received the pre-reply generate a random challenge (challenge2), encrypts challenge1 with n_i's public key and sends the encrypted challenge along with challenge2 to n_i. When n_i receives this message, it encrypts challenge2 with n_j's public key and sends the route reply along with the encrypted value of challenge2 to n_i. This procedure is designed for detecting nodes which attempt to impersonate other nodes.

Papadimitratos and Haas presented secure routing protocol (SRP) [91]. SRP assumes the existence of a security association between a node initiating a route request query and the sought destination. The basic operation is as follows: A source node S initiates a route discovery by constructing and broadcasting a route request packet containing a source and destination address, a query sequence number, a random query identifier, a route record field (for accumulating the traversed intermediate nodes) and the message integrity codes (MIC) of the random query identifier, computed using HMAC [75] and the secret key shared between the S and the destination. Intermediate nodes relay the route request packet so that one or more query packet(s) arrive(s) at the destination. When the route requests reach the destination D, D verifies that (a) the MIC is indeed that of the random query identifier, and (b) the sequence number is equal to or greater than the last known sequence number from S. If both (a) and (b) hold, D constructs a corresponding route reply packet containing the source, destination, the accumulated route in the route record field of the request query, the sequence number, the random query identifier and the computed MIC of the above. D then sends the route reply to S using the reverse path in the route record field. When S receives a route reply packet it validates the info it contains and verifies the computed MIC. If all is well, it uses the ascertained route to communicate with D.

Hu, Johnson and Perrig proposed the Secure Efficient Ad hoc Distance vector routing protocol (SEAD) [47]. SEAD is a secure proactive protocol which is based on the design of DSDV [95]. SEAD uses one-way hash chains [76] for authenticating the hop count values in advertised routes and routing updates. For the authentication of the sender of routing update messages, SEAD allows authentication to be done using broadcast authentication mechanisms such as

TESLA [97], HORS [100] or TIK [48] which require the network nodes to have time synchronized clocks. Alternatively, SEAD allows message authentication codes to be used to authenticate the sender of routing update messages; however, this is based on the assumption that shared secret keys are established among each pair of nodes.

Zapata presented Secure AODV (SAODV) [127, 128, 126]. SAODV uses two mechanisms to secure AODV: digital signatures to authenticate non-mutable fields of the routing control messages and one-way hash chains (as is the case for SEAD, outlined above) to secure hop count information.

Hu, Perrig and Johnson proposed a routing security scheme called Ariadne [46] which is based on the design of DSR [61]. Ariadne uses message authentication code for authenticating routing control messages, and it requires time synchronization hardware for synchronizing the release of the secret keys used for generating the message authentication codes.

Sanzgiri and Dahill presented ARAN [105]. ARAN uses digital certificates to secure the routing control messages. In ARAN route discovery phase, a source node S constructs a route discovery packet (RDP), signs it, attaches its certificate and broadcasts it to its neighbors. When a node A, which is a neighbor of S, receives the RDP message, if it has not previously seen this message, it verifies the signature using the attached certificate, signs the RDP message, attaches its certificate and broadcasts it to its neighbors. An intermediate node B which is a neighbor of A, on receiving the RDP message, it validates the signature using the attached certificate. B then removes A's certificate and signature, records B as its predecessor, signs the message and broadcasts it to its neighbors. The process continues in this manner until a RDP message arrives at the destination D. D selects the first RDP message it received, uses it to construct a reply (REP) packet and unicasts it to S using the reverse path. Each node on the reverse path back to S validates its predecessor signature using the attached certificate, removes the signature and the certificate (if the certificate does not belong to the destination node D), signs the packet, attaches its certificate and forwards the packet to the next-hop. Eventually, S should receive the REP with the route it seeks.

Hu, Perrig and Johnson presented a mechanism called packet leashes for detecting and defending against wormhole attacks [48]. In wormhole attacks, an attacker receives packets at one point in a network, tunnel them to another point in the network and replays them into the network from that point. The authors proposed two types of packet leashes: geographical leashes and temporal leashes. Geographical leashes require a node to know its own geographical location and all nodes must have loosely synchronized clocks. Whereas temporal leashes require all nodes to have tightly synchronized clocks. The leash mechanisms add necessary fields to a packet—for example the time the packet was sent and the sender's geographical location (for geographical leashes)—which allows the receivers to validate whether a node is in its transmission range or not. The authors also proposed a secure broadcast scheme called TIK which can be used to secure the packet leash mechanisms.

3.2.2 Trust-based routing schemes

The routing security schemes which fall in this category assign quantitative or qualitative trust values to the nodes in the network, based on observed behavior of the nodes in question. The trust values are then used as additional metrics for the routing protocols. We commence the review with one of the earlier protocols.

Yi *et al* proposed a scheme called security-aware ad hoc routing (SAR) [124]. In SAR, nodes are categorized based on their security level. A secret group key is associated with each security level and it is shared amongst nodes which are classified at the given security level. SAR incorporates security attributes as route discovery parameters, such that a node can specify its preference with regards to the security level required for participation in the routing process.

Yan, Zhang and Virtanen proposed a trust evaluation based security solution [122]. The application of this scheme to MANET routing is similar in principle to the design of SAR [124], in that the trust (or reputation) of a node is used as a routing metric when deciding the next hop of a packet.

Pirzada and McDonald presented a model for trust-based communication in ad hoc networks [98]. In this model, each node passively observe other nodes and assigns quantitative values (which range from 0 to +1) to nodes based on observed behavior. The authors proposed an extension of DSR [61] which incorporates the trust model and utilizes trust as an additional routing metric.

Nekkanti and Lee presented a trust based adaptive on demand routing protocol [88]. The authors articulated that the most effective way of preventing certain routing attacks is to totally hide certain routing information from unauthorized nodes. In this regard, the main aim of their proposed scheme is to mask the routing path between a source and a destination from all other node. The scheme is based on AODV [96]. It stipulates that one of three possible encryption levels be applied to a route request packets (RREQ). The encryption levels are high encryption which requires a 128-bit key, low encryption which needs a 32-bit key, and no encryption. The security level of a node and the security level of an application determine which encryption level is utilized. The general idea is that the more trustworthy a node is, the less need there is to hide routing information from this node during a route discovery operation. A summary of the route discovery operation is as follows: A source node S which desires a route to a destination D constructs a RREQ packet. The RREQ has a field where the application can set the security level it requires. The source then utilizes the public key of the destination node D to encrypt (with the appropriate security level) the source ID field of the RREQ packet and broadcasts it to its neighbors. When an intermediate node receives a RREQ packet it has not previously seen, if it is not the destination, it adds its node ID to the packet, signs it then encrypts it using the the public key of D and broadcasts it to its neighbor. Eventually an RREQ packet should get to D. On receiving an RREQ packet, D verifies the signatures, decrypts the encrypted fields and verifies that the nodes in the path has the minimum required trust level. If these validation operations succeed, it constructs a route reply (RREP) packet and a flow-id and encrypts the RREP and the flow-id with the public keys of the nodes in the reverse path to S (in the order that the nodes should receive the RREP packet); then D signs the encrypted RREP and broadcasts it to its neighbors. When an intermediate node n_i receives the RREP it will attempt to decrypt it; if the decryption operation fails, n_i discards the packet; otherwise, it updates its routing table, removes its part of the RREP and broadcasts it to its neighbor. Eventually, the RREP should get to the source S which will verify the signature and decrypts the RREP to ascertain the route it seeks.

Boukerche *et al* proposed secure distributed anonymous routing protocol (SDAR) [16]. The main objective of SDAR is to allow trustworthy intermediate nodes to participate in routing without compromising their anonymity. SDAR utilizes a trust management system which assigns trust values to nodes based on observed behavior of the nodes, along with recommendation from other nodes. SDAR requires each node to construct two symmetric keys, and shares one with its neighbors which have high trust values, and the other with its neighbors which have medium trust values. When a node S desires to discover a routing path to a destination D, S constructs a routing request packet (RREQ), part of which is un-encrypted and the other part encrypted. The un-encrypted part of the RREQ contains necessary routing information such as the trust level requirement of the message and a one-time public key TPK. The encrypted part of the RREQ packet contains the destination ID, a symmetric key K_s generated by S and the private key TSK for the one-time public key TPK, plus other information. Part of the encrypted portion of the message is encrypted with the public key for the destination D and the other portion is encrypted with the symmetric key K_s. S then encrypts the entire packet with the shared key for the appropriate security level of the message and broadcasts it to its neighbors. When an intermediate node

n_i receives the RREQ packet, it discards the message if it is not able to decrypt it. If n_i succeeds in decrypting the message, n_i adds its ID and a session key K_i then signs the portion it added and encrypts it with the one-time public TPK embedded in the un-encrypted portion of the RREQ packet; n_i then encrypts the entire message with the key (of the appropriate security) it shares with it neighbors and broadcasts the message. Eventually the message should get to D which decrypts the message with the appropriate keys. After verifying the signatures, D constructs a route reply (RREP) and encrypts it, first using the symmetric key K_s S attached, then encrypts it again using the session keys K_i's in the order that the corresponding intermediate node should receive the RREP packet. D then forwards the RREP to its neighbor. The neighbor which is the intended next-hop will decrypt its portion of the packet and forwards it to its neighbors (one of which will be able to partly decrypt it). The process continues until the RREP gets to the source node S which will be able to decrypt the entire packet and ascertain the route it seeks.

Li and Singhal proposed a secure routing scheme [78] which utilizes recommendation and trust evaluation to establish trust relationships between network entities. The scheme uses a distributed authentication model which operates as follows: each network node maintains a trust table which assigns a quantitative trust value to known network entities. If a node S desires to know the trust value of a node n_i and n_i is not in S trust table, S sends out a trust query message—to ascertain n_i's trust value—to all the trustworthy nodes in S trust table. When a node n_j receives the trust query message, if n_i is in its trust table, it sends the indicated trust value to S; otherwise it sends out a trust query message—requesting the trust value of n_i—to all the trustworthy nodes in its trust table. The process continues recursively until eventually a node which has n_i in its trust table forwards the trust value to the node which requested the info, which will in turn forward it to the node which sent it the trust query message; and so on, until eventually the response gets to S. S consequently uses the responses to compute a trust value for the node in question. This distributed authentication model is used to determine the trustworthiness of the network nodes. The end result being that nodes which are considered untrustworthy are excluded from routing paths.

3.2.3 Incentive-base schemes

In this section we present a brief description of proposed schemes which attempt to stimulate cooperation among selfish nodes by providing incentives to the network nodes.

Buttyán and Hubaux proposed an incentive-based system for stimulating cooperation in MANETs [18]. The scheme requires each network node to have a tamper resistant hardware module, called security module. The security module maintains a counter, called nuglet counter, which decreases when a node sends a packet as originator, and increases when a node forwards a packet. The operation of the scheme is as follows: when a node S desires to send a packet to a destination D, if the number of intermediate nodes on the path from S to D is n, then S's nuglet counter must be greater than or equal to n in order for S to send the packet. If S has enough nuglets to send the packet, S decreases its nuglet counter by n after sending the packet. On the other hand, S increases its nuglet counter by one each time S forwards a packet on behalf of other nodes. The value of a nuglet counter must be positive; therefore, it is within a node's interest to forward packets on behalf of other nodes, and refrain from sending large number of packets to distant destinations.

Zhong, Chen and Yang presented Sprite: A Simple, Cheat-Proof, Credit-Based System for MANETs [130]. Sprite provides incentive for MANET nodes to cooperate and report actions honestly. Sprite requires a centralized entity called a Credit Clearance Service (CCS) which determines the charge and credit involve in sending a message. The basic operation of Sprite is as follows: when a node receives a message, the node keeps a receipt of the message. Later when the node has a fast connection to a CCS, it reports to the CCS the message it has received/forwarded by uploading its receipt. The CCS then uses the receipt to determine the charge and credit involve in the transmission of the message.

3.2.4 Schemes which employ detection and isolation mechanisms

This section contains a brief description of schemes which utilize detection and isolation techniques. We commence the review with an earlier proposal.

Marti *et al* [82] proposed a scheme for mitigating against the presence of MANETs nodes that agree to forward packet but fail to do so. The scheme utilizes a "watchdog" for identifying misbehaving nodes and a "pathrater" for avoiding those nodes. Each node has its own watchdog and pathrater modules. Watchdog operation requires the nodes within a MANET to operate in promiscuous mode: meaning that a node n_i that is within the transmission range of a node n_j should be able to overhear communications to and from n_j even if those communications do not involve n_i. Watchdog is based on the assumption that if a packet was transmitted to node n_i for it to forward the packet to node n_j, and a neighboring node to n_i does not hear the transmission going from n_i to n_j then it is likely that n_i is malicious and should therefore be assigned a lower rating. Pathrater is responsible of assigning ratings. The rating is assigned as follows: when a node n_i become known to the pathrater, n_i is assigned a "neutral" rating of 0.5. The ratings of nodes which are on actively used path are consequently incremented by 0.01 every 200 ms; whereas, a node's rating is decremented by 0.05 when a link to the node is surmised to be nonfunctional. "Neutral" ratings are bounded with an upper bound of 0.8 and a lower bound of 0.0; but a node always assign a rating of 1.0 to itself. Rather than selecting a path to a given destination based on the number of hops in the path, the pathrater selects the path which has the highest average rating.

Buchegger and Le Boudec proposed a protocol called CONFIDANT [104] that aims to detect and isolate misbehaving nodes in MANETs. CONFIDANT uses a form of reputation systems [99] where the nodes within a MANET rate each other based on observed behaviors. Nodes that are deemed to be misbehaving are placed on black lists and are consequently isolated.

Awerbuch *et al* presented a routing security scheme [6] aimed at providing resilience to byzantine failure caused by individual or colluding MANET nodes. The scheme utilizes digital signature for authentication at each hop, and it requires each node to maintain a weight list consisting of the reliability metric of the nodes within the network. The weight list is used in the route discovery phase to avoid faulty paths. When faults are detected in established paths, an adaptive probing technique is launched in an attempt to detect the faulty links. Faulty links are given decreased rating and are consequently avoided.

Just and Kranakis [63] and Kargl *et al* [65] proposed schemes for detecting selfish or malicious nodes in an ad hoc network. The schemes involve probing mechanisms which are similar in functionality to that of Awerbuch *et al* [6] above.

Patwardhan and Iorga [93] presented a secure routing protocol called SecAODV. SecAODV is based on AODV but unlike the latter, it requires each node in the MANET to have a static IPv6 address. The scheme allows source and destination nodes to establish secure communication channel based on the concept of Statistically Unique and Cryptographically Verifiable (SUCV) identifiers [83] which ensures secure binding between an IPv6 address and a key, without requiring any trusted certificate authority (CA). SecAODV also provides an IDS (intrusion detection system) for monitoring the nodes' activities.

CHAPTER 4
Motivation for a decentralized MANET certificate revocation scheme

The issue of certificate revocation in Mobile ad hoc networks (MANETs) where there are no on-line access to trusted authorities, is a challenging problem. In wired network environments, when certificates are to be revoked, certificate authorities (CAs) add the information regarding the certificates in question to certificate revocation lists (CRLs) and post the CRLs on accessible repositories or distribute them to relevant entities. In purely ad hoc networks, there are typically no access to centralized repositories or trusted authorities; therefore the conventional method of certificate revocation is not applicable.

In this thesis, we present a decentralized certificate revocation scheme that allows the nodes within a MANET to revoke the certificates of malicious entities. The scheme is fully contained and it does not rely on inputs from centralized or external entities. Preliminary results of this research project has been published in [27, 29] and the final results [4] are accepted for publication in Elsevier Ad Hoc Networks Journal.

4.1 Analysis of existing MANET security schemes

As MANETs become more ubiquitous, the need for adequate security in these networks is more evident. Security schemes for MANETs generally employ one or more of the following cryptographic technologies: symmetric-key cryptography, digital certificates or threshold cryptography. Each of these cryptographic tools has its particular advantages and drawbacks. We address these issues in the respective subsections below.

4.1.1 Schemes based solely on symmetric-key cryptography

Security schemes involving symmetric-key cryptography are much less computationally exhaustive than those involving digital certificates or threshold cryptography. Consequently, the use of symmetric-key cryptography has much smaller computational overhead than that associated with digital certificates or threshold cryptography. However, security schemes which are based solely on symmetric-key cryptography are less robust and offer lower degree of security than those involving asymmetric key cryptography, owing to the following:

- *Greater probability of shared key being compromised*: If a secret key k is shared among a network consisting of n hosts, the probability of the key being discovered, increases proportionally with n. Therefore, for optimal security, it is necessary for k to be changed at high frequency.
- *If a single host is compromised, the entire network can be compromised*: The discovery of the secret key k on a single host, means that this key will need to be discarded and a new key distributed to all the host that shared it. If there are no key management mechanisms in place, the keys would need to be distributed through secure out-of-band means. This could be rather time consuming and problematic for medium or large scale networks.
- *Scalability issues*: As outlined above, if a secret key is shared amongst a group of hosts, it is necessary that the key be changed periodically; the

frequency depends on the level of security desired. For protocol such as IEEE 802.11 related standards (WEP, TKIP and CCMP) [54, 57], Stajano and Anderson [110] and Balfanz *et al* [7] schemes, the keys need to be distributed by secure out-of-band means. This might not be an issue for small networks; however, this task could be quite tedious and problematic for larger networks, and is therefore not a scalable solution.

As outlined in Section 2.1.1, TKIP and CCMP (IEEE 802.11i security mechanisms) have optional key management facilities which use IEEE 802.1X [55] authentication protocol. However, the IEEE 802.1X authentication protocol requires access to centralized repositories which may not be available in purely ad hoc network environments. Consequently, TKIP and CCMP key management framework via IEEE 802.1X authentication protocol is not viable in purely ad hoc networks owing to the requirement of on-line access to centralized entities.

4.1.2 Schemes involving digital certificates

Digital certificates are important elements in most commonly used network security applications, particularly those providing authentication services. Perhaps the single feature that accounts for the attractiveness of digital certificate technology is the key management issues it favorably addresses, as summarized below:

- *Simplify key distribution*: Digital certificates do not need to be kept private. There are therefore no need for secure channels for distributing certificates.

- *Reduce effect of compromise*: The fact that the certificates are not shared by entities, if the private key associated with a given certificate is compromised– unlike the case of shared secret key technology, which necessitate the issuing of a new key to all the entities sharing the key–in most cases, it suffices to replace only the certificate whose associated private key has been compromised. The exception is, if it is the CA key that is compromised, then it might be necessary to revoked all the certificates previously issued by that CA. The more stringent security measures applied to CAs private keys, should however reduce the likelihood of they being compromised.

Certificates issued via non-threshold cryptographic schemes require the utilization of some sort of trust model. The most commonly used trust models are (a) hierarchical and (b) web-of-trust models. The hierarchical trust model is the more structured approach and the most widely used. In the hierarchical trust model, a root certificate authority (CA) issues certificates to delegated CAs or end users, the CAs in turn issue certificates to end users or to other CAs. Fig. 4–1 illustrates the hierarchical trust model. The PKI X.509 (PKIX) framework [24] exemplifies this trust model.

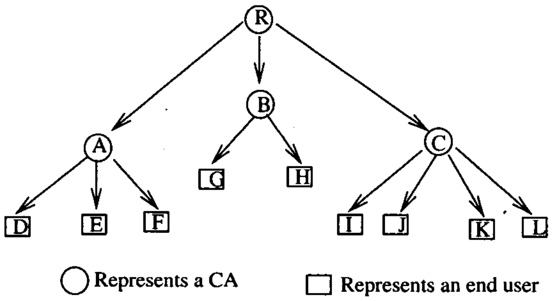

Figure 4–1: Hierarchical trust model

The web-of-trust model [133] is the more distributed approach. In this model, there is no distinction between CAs and end users. End users are responsible for all certificate management tasks, such as issuing, storage and revocation of certificates. An end user A issues a certificate to another user B if A trusts B or if a user C that A trusts, vouches for B. Fig. 4–2 illustrates the web-of-trust model. The web-of-trust model appears attractive for utilization in MANETs security schemes, owing to its distributed nature. However, the web-of-trust model is far more susceptible to infiltration of malicious agents than the more structured hierarchical model, since the latter allows much greater accountability than the former. Consider for example a network where a node A trusts another node B; if B happens to be a malicious agent, B can issue valid certificates to several other malicious agents who would be implicitly trusted by A since B—who A trusts—vouches for these agents. Similarly, if other nodes trust B, these nodes would also implicitly trust the malicious agents B vouches for. Consequently, a number of malicious agents can gain access to the network if a single untrustworthy node happens to convince another node to issue it a valid certificate.

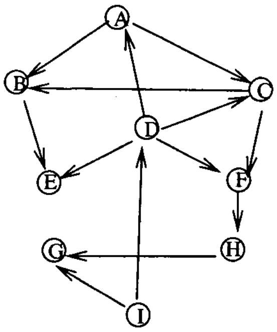

Figure 4–2: Web-of-trust trust model

The hierarchical trust model offers greater protection against this eventuality, in that the end users are accountable to the CAs that issue the certificates, and the CAs are in turn accountable to other CAs or to the root CA. If a network is compromised, this accountability structure allows the elimination of malicious agents much more readily. Hierarchical trust model or hybrid models such as that utilized in SPKI/SDSI [36, 102] and KeyNote certificates [10] (see Section 2.2) is therefore more preferable, particularly in environments where higher degrees of accountability and security are required. Security schemes such as [51, 52, 12, 20] are viable solutions for some MANET environments; however, owing to the fact that they utilize the less stringent web-of-trust model, they may not be suitable for MANETs environments where high degrees of accountability and security are required.

There are some notable challenges however in utilizing certificates that are based on the more reliable hierarchical trust model in MANETs, owing to the decentralized nature of these networks. One particular challenging problem is the issue of certificate revocation. For various reasons—such as the compromisation of private keys—certificates will need to be revoked periodically, and network peers need to be informed about the revoked certificates in a timely manner. For conventional networks, CAs issue certificate revocation lists (CRLs) [45] which contain information about revoked certificates, at regular intervals. The CRLs are then either broadcast to the relevant nodes, or placed on easily accessible centralized repositories. Alternatively, on-line certificate status protocol (OCSP) [85] can be used to ascertain information about the status of a certificate. These methodologies are not applicable to MANETs, owing to the fact that MANETs do not contain centralized entities, and they typically do not provide on-line access to external entities such as CAs.

Most of the proposed ad hoc network security schemes which utilized certificates which do not rely on web-of-trust model, do not explicitly address the issue of certificate revocation. Examples of these schemes include [114, 37, 83, 67, 68]. Other proposals such as Morogan and Muftic [84] and Verma *et al* [116] schemes make the assumption that periodic access to on-line CAs is available; therefore CRLs can be obtained from the CAs. Then there are proposals such as Candolin and Kari [19] which make provision for certificate revocation, and do not assume that on-line CAs are accessible; but they do not provide protection against certificates being wrongfully revoked through malicious accusations.

4.1.3 Threshold cryptography schemes

The utilization of threshold cryptography for the design of MANETs security schemes has generated some interest. Section 2.3.1 contains a review of MANET security schemes which employ threshold cryptography. Zhou and Haas [131] first forwarded the idea of utilizing threshold cryptography to distribute trust in ad hoc networks. They suggested that the challenges associated with key management services in ad hoc networks can be resolved by distributing CA's duties amongst the network nodes. For example, a CA signing key can be partitioned into n shares and distributed to n nodes. Any k of the n nodes could then collaborate to sign and issue valid digital certificates; whereas a coalition of $k-1$ or less nodes would not be able to do so. This allows certificates to be issued on the fly in ad hoc network environments, without input from centralized entities. Zhou and Haas scheme—as is the case for most existing MANET security schemes which involve threshold cryptography (for example [132, 69, 117, 77, 123, 121])—do not address the issue of certificate revocation.

Kong *et al* [73] presented a threshold cryptography security scheme which involves a certificate revocation mechanism. With regard to the certificate revocation mechanism, the authors suggested that if a node S considers another node n_i to be compromised, then S can generate a counter-certificate (which revokes the associated certificate), have it signed via threshold cryptography techniques and broadcasts it over the network. This certificate revocation mechanism is susceptible to malicious accusation exploits since a single malicious node can cause the revocation of another node's certificate.

Luo *et al* [80, 72] presented extensions of Kong *et al* work [73]. These proposals involve an improved certificate revocation mechanism (see Section 2.3.1) which provides some measure of assurance against certificates being wrongfully revoked through malicious accusations; however, threshold cryptographic schemes have the following noticeable drawbacks:

- *Computationally exhaustive*: As indicated in Section 2.3, threshold cryptography involves additional computationally intensive modular exponentiations compared to the underlined asymmetric-key cryptographic protocols. Most low powered wireless nodes do not have the resources to handle such computationally intensive operations. For nodes with less resources constraint, the increase in latency due to the extra computational cost, may not be acceptable. For example, the analysis of the implementation in Luo *et al* scheme [80] indicates that generation of a partial RSA signature using one of k shares, is approximately 2.5 times slower than standard RSA signing. Considering that k partial signatures needs to be generated then combined to obtain a valid signature, the increase in latency due to the additional computation may not be acceptable. Luo *et al* [80] states that shares update took approximately 80 seconds to complete on low-end devices they employed for testing. It is noteworthy to mention that the update scheme they utilized did not entail verifiable secret sharing (VSS); therefore, it only provides protection against passive adversaries. For protection against active adversaries capable of destroying shares by misbehaving in share update schemes, VSS is necessary. Since VSS based share renewal schemes involve additional modular exponentiations compared to those schemes based on simple Shamir secret sharing, we expect greater computational delays than that reported, if the share update scheme utilized in [80], provided protection against active adversaries.

- *Require unselfish cooperation*: Network security solutions involving threshold cryptography require unselfish cooperation of the communicating peers. This might not be an issue in certain military applications; however, in most commercial network applications, nodes may not behave unselfishly. Wireless nodes are often limited in battery power and utilize power conservation mechanisms that encourage them to remain dormant unless they are performing necessary services. It might not be realistic therefore to expect nodes in certain environments to behave unselfishly and cooperate, for example to service certificate requests.

In additional to the above, most of the existing MANET security proposals which involve threshold cryptography—including Luo *et al* [80] and Kong *et al* [73] schemes—are susceptible to Sybil attack [34], where nodes can spoof their identities and acquire multiple certificates.

The issue of certificate revocation in MANETs is therefore still considered as an open problem.

4.2 Reputation systems

One of the major contributions of this thesis is a localized certificate revocation scheme we developed. The scheme uses a reputation system which assigns quantitative weights to the nodes in a MANET, based on the behavior profiles of the nodes. We provide arguments below to support our claim that existing reputation systems are not applicable to certificate revocation schemes.

A number of reputation systems have been published in research literature. These systems can be divided into two main types: centralized and distributed reputation systems. Centralized reputation systems require central authorities for collecting the rating of participants and derive reputation scores. Examples of these systems are [99, 106]: the reputation systems on which eBay[1] forum and

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

Amazon[2] , respectively, are based; and the page ranking scheme [90] developed by the founders of Google[3] . Centralized reputation systems are not suitable for MANETs since MANETs do not have centralized entities. Decentralized systems are more fitting for MANET applications. The majority of proposed decentralized reputation systems are transactional based; that is, they require inputs—such as size of upload or down files, quality, price and upload/download experiences—relating to interactions of providers of services and users of the services. Examples of transactional based reputation systems are [64, 120, 42, 79, 53]. The non-transactional based systems previously proposed are not suitable for application in certificate revocation schemes because they are either too complex and have high associated overhead [125, 2], or they are based on assumptions such as those outlined in [3, 129], which are not applicable to certificate revocation schemes.

[2] http://www.amazon.com

[3] http://www.google.com

CHAPTER 5
Motivation for a secure MANET routing protocol for adversarial environments

Secure routing in mobile ad hoc networks (MANETs) has emerged as a important MANET research area. MANETs, by virtue of the fact that they are wireless networks, are more vulnerable to intrusion by malicious agents than wired networks. In wired networks, appropriate physical security measures, such as restriction of physical access to network infrastructures, can be used to attenuate the risk of intrusions. Physical security measures are less effective however in limiting access to wireless network mediums. Consequently, MANETs are much more susceptible to infiltration by malicious agents. Authentication mechanisms can help to prevent unauthorized access to MANETs. However, considering the high likelihood that nodes with proper authentication credentials can be taken over by malicious entities, there are needs for security protocols which allow MANET nodes to operate in potential adversarial environments.

In this thesis, we present a secure on-demand MANET routing protocol, we named Robust Source Routing (RSR). In addition to providing data origin authentication services and integrity checks, RSR is able to mitigate against intelligent, colluding malicious agents which selectively drop or modify packets they agreed to forward. Simulation studies confirm that RSR is capable of maintaining high delivery ratio even when a majority of the MANET nodes are malicious.

5.1 Analysis of existing MANET secure routing schemes

Research have shown that misbehaving nodes in a MANET can adversely affect the availability of services in the network [82]. Nodes misbehave either because they are broken, selfish or malicious. Broken nodes are non-functional. A node can agree to forward traffic on behalf of other nodes but becomes non-functional prior to it fulfilling this agreement. Selfish nodes can agree to forward packets but silently drop the packets in attempt to conserve energy and bandwidth. Malicious nodes may seek to disrupt a network and hide their malicious behavior by selectively dropping packets they agreed to forward. They may also attempt to create denial of service exploits by injecting large number of packets into the network. Most of the existing MANET secure routing schemes, for example [9, 115, 91, 47, 127, 61, 105], do not mitigate against these misbehaviors.

The existing schemes which attempt to mitigate against these misbehaviors use three main approaches: trust-based routing, incentive-based schemes, and schemes employing detection and isolation mechanisms. We reviewed the schemes which fall in these categories in Sections 3.2.2, 3.2.3 and 3.2.4 respectively. In this chapter, we analyze these schemes and highlight the short comings which necessitate the needs for a more robust secure routing protocol.

5.1.1 Trust-based routing

Yi *et al* proposed SAR (security-aware ad hoc routing) [124]. SAR classifies nodes based on their trust level. Nodes which have the same classification share a secret group key. In a route discovery process, the source node S can stipulate the minimum security requirement a node must have in order to be an element in the routing path from S to a destination D. S can enforce the stipulation by encrypting the route request packet with the shared key associated with the specified security level. This approach has its virtues; however, key sharing can be problematic: considering the possibility that malicious agents can take over nodes with high security classifications and gain access to the secret group keys.

Yan, Zhang and Virtanen proposed a trust model which assigns quantitative trust values to nodes based on observed behavior of the nodes [122]. The application of this trust evaluation mechanism in routing schemes is similar in principle to SAR [124]. Unlike SAR though, Yan *et al* proposal does not suggest a means whereby a source node S can prevent a node—which does not meet the trust level requirement—from being on a routing path from S to a given destination.

Pirzada and McDonald presented a model for trust-based communication in ad hoc networks [98]. The trust model depends on features such as passive or active acknowledgment of packets, gratuitous route replies (recommendations from other nodes regarding possible shorter routes) and routing error information. This scheme is susceptible to malicious accusation attacks in that malicious nodes can selectively drop packets and wrongfully accuse well-behaving nodes of misbehavior.

Nekkanti and Lee proposed a trust based adaptive on-demand routing protocol [88]. The protocol uses encryption mechanisms to mask the routing path between the source and destination from all the other nodes. This scheme provides a degree of anonymity for nodes in routing paths; but it does not provide protection against misbehaving nodes which selectively drop packets they agreed to forward.

Boukerche *et al* proposed SDAR (secure distributed anonymous routing protocol) [16]. The main objective of SDAR is to allow nodes to participate in routing without compromising their anonymity. The authors suggested that as a means of countering malicious dropping behavior, nodes can operate their network interfaces in promiscuous mode[1] and report observed discrepancies regarding unconfirmed packet transmission. This operation is similar to that of Marti *et al* [82] "Watchdog" operation and is therefore susceptible to the short comings—outline in Section 5.1.3 below—associated with Marti *et al* scheme.

Li and Singhal proposed a secure routing scheme [78] which utilizes observed behavior patterns and recommendations from other nodes to assign quantitative trust values to the nodes in a MANET. The scheme has its merits but malicious agents can thwart the scheme by dropping the trust query messages, and in so doing, renders the scheme ineffective.

5.1.2 Incentive-based schemes

Buttyán and Hubaux proposed an incentive-based system for stimulating cooperation in MANETs [18]. The scheme requires each network node to have a tamper resistant hardware module, called security module. The security module

[1] meaning that a node n_i which is within the transmission range of a node n_j should be able to overhear communications to and from n_j even if those communications do not involve n_i

maintains a counter, referred to as nuglet counter, which decreases when a node sends a packet as originator, and increases when a node forwards a packet. The scheme stipulates that each node's nuglet counter must remain positive; consequently, nodes are encouraged to forward packets for other nodes and refrain from sending large number of packets to distant destinations. The scheme offers an effective mechanism for discouraging selfishness; however it may not experience widespread use because of the requirement for a tamper resistant hard module.

Zhong, Chen and Yang presented Sprite: A Simple, Cheat-Proof, Credit-Based System for MANETs [130]. Sprite provides incentive for MANET nodes to cooperate and report actions honestly. It avoids the requirement of tamper resistant hardware module; instead, it requires on-line access to a centralized entity called a Credit Clearance Service (CCS), which determines the charge and credit involved in transmitting a message. This scheme is based on the assumption that on-line access to a CCS is available. This assumption may not hold for purely ad hoc networks, which do not guarantee access to on-line entities.

5.1.3 Schemes employing detection and isolation mechanisms

Marti *et al* [82] proposed a scheme for mitigating against the presence of MANET nodes that agree to forward packet but fail to do so. The scheme utilizes a "watchdog" for identifying misbehaving nodes and a "pathrater" for avoiding those nodes. Watchdog operation requires the nodes within a MANET to operate in promiscuous mode. Watchdog is based on the assumption that if a packet was transmitted to node n_i for it to forward the packet to node n_j, and a neighboring node to n_i does not hear the transmission going from n_i to n_j then it is likely that n_i is malicious and should therefore be assigned a lower rating. This scheme has several weaknesses. As described in the authors' own words: "Watchdog's weakness are that it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping."

Buchegger and Le Boudec proposed a protocol called CONFIDANT [104] that aims to detect and isolate misbehaving nodes in MANETs. CONFIDANT uses a form of reputation systems [99] where the nodes within a MANET rate each other based on observed behaviors. Nodes that are deemed to be misbehaving are placed on black lists and are consequently isolated. The reputation systems, however, do not provide any protection against false accusations. Consequently, the scheme is susceptible to blackmailing.

Awerbuch *et al* presented a routing security scheme [6] aimed at providing resilience to byzantine failure caused by individual or colluding MANET nodes. The scheme utilizes digital signature for authentication at each hop, and it requires each node to maintain a weight list consisting of reliability metrics of the nodes within the network. The weight list is used in the route discovery phase to avoid faulty paths. When faults are detected in established paths, an adaptive probing technique is launched in an attempt to detect the faulty links. Faulty links are given decreased rating and are consequently avoided. Probing techniques are useful in identifying faults caused by non-malicious acts. However, they are ineffective against malicious agents, simply because the probing packets are distinguishable from other packets; therefore, an adversary can choose to behave well when it is being probed, but behave maliciously during intervals when it is not being probed.

Just and Kranakis [63] and Kargl *et al* [65] proposed schemes for detecting selfish or malicious nodes in an ad hoc networks. The schemes involve probing mechanisms which as is the case with [6], the probing packets are distinguishable from other packets.

Patwardhan and Iorga [93] presented a secure routing protocol called SecAODV. SecAODV is based on AODV but unlike the latter, it requires each node in the MANET to have a static IPv6 address. The scheme allows source and destination nodes to establish secure communication channels based on the concept of Statistically Unique and Cryptographically Verifiable (SUCV) identifiers [83] which ensures secure binding between an IPv6 address and a key, without requiring any trusted certificate authority (CA). The application of this protocol is currently very restrictive because of the requirement that each of the MANET nodes must have a static IPv6 address.

Summary

Table 5–1 summaries the analysis we presented in this chapter. The analysis shows that the existing secure MANET routing schemes do not adequately mitigate against misbehaving nodes which selectively drop packets they agreed to forward, and in so doing, these misbehaving nodes can cause various network communication problems. The secure routing protocol we developed is aimed at addressing this security need.

Table 5–1: Summary of routing security analysis

Schemes	Comments
Schemes which do not address packet dropping	SRP [91], SEAD [47], SAODV [127], Ariadne [46], ARAN [105], Binkley *et al* [9] and Venkatraman *et al* [115] schemes do not address packet dropping.
Trust-based schemes	SAR [124] requires shared group keys; therefore it is subjected to the key management issues outlined in Section 4.1.1. Pirzada *et al* and Nekkanti *et al* [98, 88] do not provide protection against packet dropping; SDAR [16] is subjected to the short comings indicated below for Marti *et al* scheme; Li *et al* [78] scheme can be thwarted by dropping the trust query messages.
Incentive-based schemes	Buttyán *et al* [18] requires tamper resistant hardware and Zhong *et al* [130] requires on-line access to a centralized entity; therefore, these schemes are limited in their applications.
Schemes which employ detection and isolation mechanisms	Marti *et al* [82], in the author's own words, has the following weaknesses: "it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping." Buchegger *et al* [104] scheme does not provide protection against false accusations. The probing technique Awerbuch *et al*, Just *et al* and Patwardhan *et al* schemes [6, 63, 93] utilize, is ineffective against intelligent adversaries which selectively drop packets, since the probing packets are not completely indistinguishable from other data packets.

CHAPTER 6
A localized certificate revocation scheme for MANETs

In this chapter, we present the MANET certificate revocation scheme we developed.

6.1 Overview of the certificate revocation scheme

Our scheme stipulates that before entering a network, the MANET nodes must have a valid certificate from a recognized CA, as well as the public keys of the CAs which issued certificates for potential network peers. The certificates can be used for network authentication. The nodes will be able to verify the validity of the certificates, since they have the public keys of the CAs which issued them. The MANET nodes are therefore responsible for all key management tasks except the issuing of certificates. For optimum security, a CA should verify the identity of a node before issuing it a certificate.

Our certificate revocation scheme requires the nodes in a MANET to monitor the behavior of the other nodes. If a node surmises that a given node is behaving suspiciously, it is required to broadcast an accusation against the node in question. Our scheme utilizes the self-healing community approach presented in [71] for disseminating the accusation information via broadcast. Self-healing community approach is based on the observation that in a MANET, any node that is within both node A and node C transmission range can in principle forward packets from node A to C. For example, in Fig. 6–1, node A and C are outside the transmission range of each other. In principle, any of the nodes $(n1, n2, n3, n4)$ within the self-healing community can forward packet from A to C. So, if a malicious or selfish node within a self-healing community chooses not to forward a packet it is asked to forward, any other node within the community can provide the service instead. A self-healing community is functional as long as there is at least one well-behaving node in the community. This approach requires the network interfaces of the MANET nodes to stay in promiscuous reception mode. For further detail and analysis of the self-healing community concept, see [71].

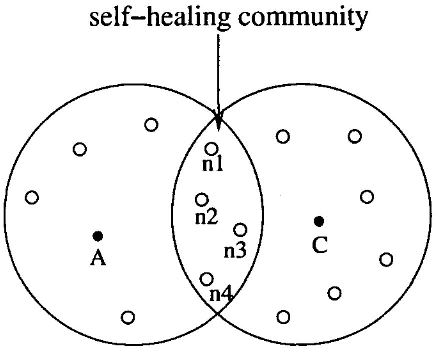

Figure 6–1: Self-healing community forwarding

Our certificate revocation scheme requires each participating node to compile and maintain data—based on broadcast accusation information—about all the nodes in the network. The collected data is used to assign a quantitative value for the trustworthiness of a node. Accusations from any given node are weighted based on the trustworthiness of the accuser: the higher the trustworthiness of a node, the greater the weight of its accusations, and vice versa. A node's certificate is revoked if the value of the sum of accusation weights against the given node is greater than a configurable threshold. The protocol aims at providing similar data to each node for computing the trust ratings of the network peers; the end goal being that the nodes have consistent information regarding the status of the certificates of their network peers.

6.1.1 Cryptographic primitives

For efficiency considerations, rather than relying on digital signatures for message origin authentication and content integrity checks, we mainly use one-way hash chains [76]. One-way hash chains are based on one-way hash functions. A one-way hash function H, maps an input x of any length to an output y of fixed length, such that, given y, it is computationally infeasible to find x, where $H(x) = y$. Two commonly used one-way hash functions are SHA-1 [89]—which produces 160-bit outputs—and MD5 [101], which gives 128-bit outputs.

A one-way hash chain can be created by choosing a random value x of arbitrary length and compute the hash chain values $y_0, y_1, y_2, ..., y_{n-1}, y_n$, where $y_0 = x$ and $y_i = H(y_{i-1})$, such that $0 < i \leq n$, for a given n. The hash chain values—in order of decreasing subscript i (that is, from right to left in the list above)—at varying point in time can then be used for authentication or as symmetric keys for keyed hashing functions such as HMAC [75]. When the hash chain values are used as keys for keyed hashing functions, for example, y_n can be signed and be distributed to network peers who will use it to authenticate the other y_i values. y_{n-1} can then be utilized with HMAC to generate a message integrity code (MIC) for a message m_1, and appended to m_1 before it is transmitted. After a designated time period, y_{n-1} is released and utilized by the recipient of m_1 to verify the message integrity. Similarly, at a later point in time, y_{n-2} can be used to generate a MIC for another message m_2. The network peers are able to authenticate the y_i values since y_n is signed and they can verify whether $y_{i+1} = H(y_i)$, for all previously seen $i \leq n$. Unlike TESLA [97], our protocol does not require time synchronization, owing to the unique way we utilize the hash chains.

6.2 Detail of scheme

The following assumptions are made regarding to the MANETs and the nodes that constitute the networks:

- The number of malicious or selfish nodes is less than the number of well-behaving nodes.
- The network interfaces of the nodes are capable of operating in promiscuous reception mode.
- Each node has only one valid certificate.

The first duty of a node when it enters a MANET is to compute a series of hash chain values $y_0, y_1, y_2, ..., y_{n-1}, y_n$, using an agreed upon hash function H, as outlined in Section 6.1.1, if they have not been computed a priori; sign y_n and broadcast it along with its certificate to the nodes in the network. Upon receiving a signed y_n and the corresponding certificate, the nodes verify that the certificate is valid. If it is valid and it is not revoked, and the signature on the y_n value is valid, the nodes store both the certificate and y_n; sign their profile tables and their y_n values, and unicast them to the sender of the certificate. Note that if a node has already used any of its y_i values to secure messages, it will sign and send the last y_i it utilized—as its y_n value—to new entrants to the network. A profile table contains information about the behavior profile of the nodes in the MANET.

Upon receiving the profile tables with valid signatures from its network peers, a node is required to compile its own profile table which is initially based on the information contained in the profile tables it received. Transmission of profile tables to new entrants to the network is necessary in order to ensure that the newcomers have up-to-date information regarding the behavior profile of its network peers.

A profile table can be represented as a packet of varied length depending on the number of accusations launched against the nodes. The length ranges from a minimum of 80 bits—when there are no accusations—to a maximum of $97(N-2)+145$, where N is the number of nodes in the network. A profile table contains the following fields:

1. *Owner's ID*: This field is the first 32 bits of the profile table. It contains the certificate serial number of the node that compiled the profile table.
2. *Node count*: This 16-bit field contains a short integer indicating the node perspective regarding the number of nodes in the network.
3. *Peer i ID*: This is a 32-bit field containing the certificate serial number of a node that is accused of misbehavior. This field also serves the purpose of a marker: if it contains zero, it indicates the end of the profile table.
4. *Certificate status*: This field contains 1-bit flag. The bit is set if the certificate is revoked, and unset otherwise.

5. *Accusation info*: The first 32 bits of this 64-bit field contains the certificate serial number of a node that accused peer i of misbehavior. The remaining 32 bits contain the date that the accusation was made.

If field 3 does not contain zero, the profile table continues with the certificate status and accusation information fields; and if there are more than one accusers, it continues with 97-bit blocks containing information about the other accusers. Figure 6–2 illustrates the fields of a profile table.

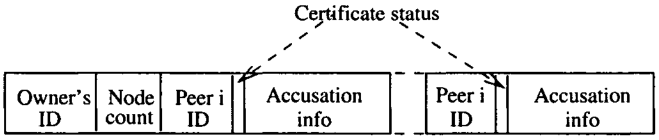

Figure 6–2: Fields of a profile table

The protocol requires each node to keep track of the following variables, the values of which are obtained from its profile table.

- *Number of accusations against node* (i) (A_i): This is the total number of accusations made against a given node i. When a node receives an authenticated accusation against node i, it updates its profile table, and consequently this variable, if and only if both node i and the accuser certificates are not revoked and no previous accusation by the accuser against node i is recorded.
- *Number of additional accusations made by node i* (α_i): When a node receives authenticated accusation information from node i, it updates its profile table and consequently this variable, if and only if the certificates of both node i and the node that is being accused of misbehavior (node j) are not revoked

and no previous accusation by node i against node j is recorded. A node is not charged for the first accusation it makes; hence, α_i is actually the total number of accusations node i made minus one.

- *Behavior index of node i* (β_i): The behavior index (β_i) of a node i is a measure of the trustworthiness of the node i. β_i is a floating point number such that $0 \leq \beta_i \leq 1$. The greater the value of β_i, the more trustworthy node i is perceived to be. β_i is computed as follows:

$$\beta_i = 1 - \lambda A_i \tag{6.1}$$

where $\lambda = \frac{1}{2N-3}$ and N is the number of nodes in the network.

- *Weight of node i accusation* (ω_i): This is a quantitative value that is assigned to the weight of a node's accusation. It depends on the behavior index of the node and on the number of accusations the node made. ω_i is a floating point number such that $0 \leq \omega_i \leq 1$. It is calculated as follows:

$$\omega_i = \beta_i - \lambda \alpha_i \tag{6.2}$$

where λ is as indicated above.

- *Revocation quotient* (R_j): This floating point number determines whether the certificate for node j should be revoked. A certificate is revoked if R_j is greater than or equal to the revocation quotient threshold R_T. R_T is a configurable parameter whose value depends on the sensitivity of the security requirement. Typical values of R_T are $\frac{1}{2}$, $\frac{1}{3}$ or $\frac{1}{4}$. R_j can be computed as follows:

$$R_j = \sum_{i=1}^{N} \sigma_{ij}\omega_i \tag{6.3}$$

where $\sigma_{ij} = 1$ if node i launched a complain against node j, and 0 otherwise.

- *Certificate status* (C_j): Indicates whether or not the certificate of node j is revoked. As indicated above, a certificate is revoked if $R_j \geq R_T$.

6.2.1 Determining the number of nodes in the network

MANETs are dynamic in nature: nodes may join and leave the networks on frequent basis. Consequently, the number of nodes N in any given MANET will likely not be constant. Our revocation scheme uses the mechanism outlined below for determining the number of nodes in the network at any given time. As outlined earlier, when a node enters a MANET, it is required to broadcast its certificate and the y_n value of its hash chain to all the network nodes. Upon receiving the broadcast, the peers are expected to unicast their certificates along with their hash chains y_n values to the new node. The certificates and the y_n values can be stored using any appropriate data structure. However, our protocol stipulates that each certificate entry should contain a field for storing an associated date. The date, including the time, that the certificate was received should initially be stored in this field.

After broadcasting its certificate, each node is required to broadcast short messages containing its certificate serial number and the date and time that the message was sent, at a configurable time interval of T minutes. The value of T depends on the frequency of the change in the network membership. We call these messages, membership confirmation messages. For message origin authentication and content integrity checks, a MIC of the message should be generated—using an agreed upon secure keyed hashing function and the hash chain value (with the highest subscript) that has not been previously used, as the key—and appended to the message. When a node receives a membership confirmation message m_i, from a node j, it stores it in memory or in a temporary file. The next membership confirmation message or accusation information message from node j, should contains the y_i value that was used to compute the MIC for the previous message (m_i) from the source. The node should first verify that the y_i value is authentic by ascertaining whether the hash of y_i equals the last previously revealed hash chain value of the source; that is, whether $y_{i+1} = H(y_i)$. If it is authentic, it computes the MIC of the message m_i using y_i as the key; if the MIC is identical to that which was appended to m_i, the node updates the date field associated with the certificate entry for node j, with the date indicated in m_i. It should be noted that, as explained in Section 6.2.2 below, the protocol does not require time synchronization.

If a node does not receive a verified authenticated membership confirmation message from any given node within $1.5T$ minutes, the certificate entry for the node in question, should be deleted from the node's certificate repository. The number of entries in the certificate repository for any given node, should therefore closely reflect the actual number of nodes in the network.

6.2.2 Security mechanism

The messages our certificate revocation protocol exchange can be categorized as follows:

1. *Initialization messages*: These messages are sent when there is a new entrant to the MANET. A new entrant broadcasts its digital certificate and its y_n value to the nodes in the network; the MANET nodes in return unicast their y_n values and profile tables to the new entrant. The protocol requires a digital signature scheme for authenticating the y_n values and the profile tables.
2. *Membership confirmation and accusation info messages*: The majority of the messages the protocol exchanges fall in this category. For efficiency considerations, we utilized hash chains for verifying the integrity and authenticity of these messages.

After a node j broadcast its certificate and its hash chain y_n value to its network peers, the next membership confirmation or accusation info message m_i it sends, it uses its hash chain y_{n-1} value to compute a MIC for m_i and appends it to m_i before sending the message. Node j then appends its y_{n-1} value to the next membership confirmation or accusation info message m_{i+1} it sends and in turn uses y_{n-2} to generate a MIC for m_{i+1}. On receiving m_i from node j, the recipients need to wait until they receive m_{i+1} from node j before they can verify the authenticity and integrity of m_i. Membership confirmation messages are sent every T minutes; T is a configurable parameter. As outlined in Section 6.2.1, an accusation messages can be sent at anytime. Therefore a node should not have to wait for more than T minutes to authenticate any given message. If a node does not receive the hash chain value required to verify the authenticity and integrity of a message m_i within $1.5T$ minutes, the node is required to discard m_i. Time synchronization is not required because the time interval T is a local parameter and as shown below in Section 6.3.1, it is not necessary to have global consensus on precisely when this interval starts or ends.

6.3 Discussion

Our certificate revocation scheme allows MANETs' nodes to revoke the certificates of malicious or misbehaving nodes; in so doing the malicious or misbehaving nodes are effectively isolated from a given MANET. The scheme is designed so as to prevent malicious nodes from being able to use wrongful accusations to cause the revocation of the certificates of well-behaving nodes. We elaborate on this issue further in Section 6.3.1.

The certificate revocation scheme provides a methodology of quantifying the trustworthiness of MANETs' nodes based on the behavior profiles of the nodes. The value of a node's trustworthiness determines the weight of its accusation. The weight of node n_i accusations, depends on the number of accusations made against node n_i, as well as the number of accusations node n_i made. If a number of accusations is made against a node, it is likely that this node in question is malicious or misbehaving. Similarly, if a node made a large number of accusations, particularly if the accusations are not supported by other nodes, it is also likely that this node is malicious. A node is not charged for the first accusation it made. Additionally, when the certificate of a node n_j is revoked, all the nodes that accused node n_j of misbehavior will have one subtracted from the individual total of the number of accusations they made. Similarly, when the certificate of a node n_j is revoked, one is subtracted from the individual total of the number of accusations against all the nodes that node n_j accused of misbehavior. In so doing, the nodes are not permanently charged for legitimate accusations they made; likewise, they are not permanently charged for accusations malicious nodes made against them. It should be noted however that when a certificate if revoked, it cannot be un-revoked. This is necessary to prevent the formation of loops in the process of deducting accusations originated from suspected malicious nodes.

The underline principle of the scheme is that the weight of a node's accusation should be exactly zero if the behavior index (trustworthiness) of the node is the minimum possible value and the node made the maximum number of accusations that is allowed. The maximum number of accusations which can be made against any given node is $N - 1$ where N is the number of nodes in the network. Therefore the minimum value for β_i is $1 - \lambda(N - 1)$. As indicated above, for fairness considerations, a node is not charged for the first accusation it made; hence the maximum number of accusations that any given node can be charged for is $N - 2$. Consequently, $\omega_i = 0$ when $A_i = N - 1$ and $\alpha_i = N - 2$, that is, $\omega_i = 1 - \lambda(N-1) - \lambda(N-2) = 0$. So the normalization variable λ, which ensures that the behavior index (β_i) is always within the range of zero and one inclusively, irrespective of the value of N, is equal to $\frac{1}{2N-3}$.

Our revocation scheme requires that new entrants to a MANET be sent the profile tables of the existing members of the MANET. This is necessary to ensure that the newcomers have up-to-date information about the behavior profile of the current members of the MANET. Unlike accusation info and membership confirmation messages, which use message integrity code (MIC) for message origin and integrity checks, profile table messages are authenticated with signatures. The use of signatures eliminate the delay in authenticating the message, in that the recipient of the profile tables do not have to wait for the release of hash chain values to authenticate the message. Profile tables are unicast only when new entrants enter a network; therefore the generation and verification of signatures for profile table messages should have minimal effect on the overall performance of the protocol.

As outlined in Section 6.1, our certificate revocation scheme utilizes the self-healing community approach presented in [71] for forwarding packets. This approach provides redundancy, in that if a malicious node drops a packet it is expected to forward, a well-behaving node in the community can detect the malicious activity and provide the service of forwarding the packet. If there is no well-behaving node in a self-healing community, adversarial agents may succeed in preventing accusation information from reaching certain nodes. Consequently there may be variations in the profile tables. In cases where there are variations, the new entrant is expected to fill the fields of its profile table with the values in the respective fields of the majority of the profile tables. This may result in differences in the computed β_i, ω_i and R_i values. Hence a certificate may not be revoked on all nodes instantaneously; however within negligible time interval, the certificate of a malicious node should be revoked on enough nodes which participate in the protocol, such that the malicious nodes will be rendered ineffective in perpetuating their adversarial behaviors.

The protocol does not require the cooperation of all nodes in a MANET. Malicious or misbehaving nodes may not adhere to the protocol; furthermore they may attempt to thwart the protocol by not forwarding accusation and membership confirmation messages. There are strong motivations though for well-behaving nodes to participate, since it is within their interest to help eliminate malicious or misbehaving nodes from the network.

6.3.1 Security analysis

In this section, we analyze the security of our certificate revocation protocol using a game-theoric approach. In the game, the goals of the adversaries are: i) to disrupt the protocol by preventing accusation information and membership confirmation messages from non-adversarial nodes from reaching their destinations; ii) prevent the revocation of their certificates; and iii) cause the revocation of certificates of well-behaving nodes. Whereas the goal of the well-behaving nodes is to revoke the certificates of malicious entities and consequently isolate them from the network. We show below that the probability of adversarial nodes achieving their goals is very low.

Security properties

If the number of well-behaving nodes (k) is sufficiently large, that is,

$k \geq \frac{2+\sqrt{4+8R_T(2N-3)}}{4}$, where R_T is the revocation quotient threshold and N is the number of nodes in the network, then the protocol is:

I) resistant to adversarial attacks;

II) effective in revoking the certificates of adversarial nodes.

Proof sketch of Property I): The proof utilizes the attack scenarios outlined below to show the following:

1) the effectiveness of the hash chain security mechanism;
2) at least R_T malicious entities are required to cause the revocation of the certificate of a well-behaving node;
3) The probability of malicious nodes succeeding in filtering messages from well-behaving nodes is very small.

1a) *As outlined in Section 6.2.2 above, there is a delay in verifying the authenticity and integrity of accusation info and membership confirmation messages because the recipients of the messages need to wait until they receive the hash chain values for computing the MIC for the given messages. One possible attack malicious nodes can mount as a result of the delay in verifying the authenticity of a message, is to delay forwarding a message m_i until it receives the message m_{i+1} which contains the key for computing the MIC for m_i; then modifies m_i and uses the key revealed in m_{i+1} to generate a new MIC for the modified m_i $(\hat{m}_i)$, appends it to $\hat{m}_i$, then forwards the modified message.*

If there are functional self-healing communities[1] , the message m_i should get to its destinations before the modified message $\hat{m}_i$. The protocol necessitates that a given y_i hash chain value cannot be used more than once. Therefore on seeing $\hat{m}_i$

[1] We outline the consequences of non-functional self-healing communities below.

been authenticated with the same hash chain value as that utilized to ascertain the authenticity of the previously received m_i, the recipient will discard the modified message $\hat{m_i}$; consequently the attack will not succeed.

1b) *Malicious nodes impersonate other nodes and use the spoofed identities to launch accusations against well-behaving nodes.*

If a malicious entity M spoofed the identity of node j, then prior to sending any accusation message using node j identity, M must prevent membership conformation and accusation messages from j from reaching well-behaving nodes. This is necessary since, as explained in item 1a) above, a hash chain value can only be used once for authenticating a message. If there are functional self-healing communities, this attack will not succeed.

2) *Adversarial entities act in collusion, target one well-behaving node at a time and launch accusations against the targeted node in efforts to cause the revocation of its certificate.*

As outlined in the heuristic argument below, this attack is only possible if the number of malicious nodes is greater than or equal to the revocation quotient threshold R_T. If we assume the worst case scenario where no accusation is made against any of the malicious nodes and the weight of the accusations (ω_i) of each of the malicious nodes is at the maximum value possible; if no accusation is made against any of the malicious nodes, then based on Equation (6.1) in Section 6.2, $\beta_i = 1$ for each of the malicious nodes; and since $\omega_i = 1$ (maximum value), then each of the malicious nodes made only one accusation, which is directed at the victim they targeted (node j). If there are m malicious nodes, based on

Equation (6.3) in Section 6.2, $R_j = m\omega_i$, that is, $R_j = m$. A certificate is revoked if $R_j \geq R_T$. Therefore if the malicious nodes are to succeed in causing the revocation of a certificate, the minimum requirement is that m must be equal to R_T. If anything other than the worst case scenario is assumed, that is, accusation(s) is/are made against any of the malicious nodes, or any of the malicious nodes made more than one accusations, then m must be greater than R_T for the malicious nodes to succeed in revoking the certificate of a well-behaving node.

3) *Adversarial entities act in collusion and create non-functional self-healing communities; consequently isolate targeted nodes from the rest of the network.*

If colluding adversarial entities form self-healing communities which contain no well-behaving node, they can essentially partition the network and isolate targeted nodes. If this occurs, the adversarial entities can reduce the effectiveness of the protocol; for example, if one or more well-behaving node(s) is/are isolated from the rest of the network, it is possible that the number of un-isolated well-behaving nodes may be less than the number of malicious nodes. If this were to occur, a key assumption on which the protocol is based would not be satisfied. It should be noted however that non-transient non-functional self-healing communities are unlikely considering that malicious nodes typically cannot restrict the movement of non-compromised nodes. Additionally, Kong *et al* [71] shows that the probability that an expected area of a self-healing community, $E(A_{heal})$, contains k honest nodes is given by:

$$\Pr[y = k] = \int\int_{E(A_{heal})} \frac{((1-\theta)\,\rho_L)^k}{k!}\, e^{-(1-\theta)\,\rho_L} dA$$

where y is a random variable for the number of honest nodes, L is the number of nodes, θ is the proportion of malicious nodes, and ρ_L is the node density function, which is dependent on the location in space. This probability function arises from a series of computations based on the spatial analytical model Kong *et al* [71] used for verifying the effectiveness of self-healing community forwarding. If $k = 0$, that is, if there are no well-behaving nodes in a self-healing community, this probability becomes

$$\Pr[y = k] = \int\int_{E(A_{heal})} e^{-(1-\theta)\,\rho_L} dA$$

which is small since the value of the function $e^{-(1-\theta)\,\rho_L}$ is small.

Hence, non-transient, non-functional self-healing communities are unlikely. Consequently, the probability of adversarial entities succeeding in filtering messages from well-behaving nodes is low; therefore, by 1a), 1b) and 2) above the protocol is resistant to adversarial attacks. □

Proof of Property II): Next, we show that the protocol is effective in revoking the certificates of malicious nodes. Recall that from 3) above, non-functional self-healing communities are unlikely.

If there are no non-functional self-healing communities, the following show that malicious entities in a MANET are incapable of preventing the revocation of their certificates provided that the number of well-behaving nodes (k) is greater than or equal to $\frac{2+\sqrt{4+8R_T(2N-3)}}{4}$, where R_T is the revocation quotient threshold and N is the number of nodes in the network. Assume the worst case scenario where each of the $N-k$ malicious nodes made an accusation against each of the k well-behaving nodes. Based on Equation (6.1) in Section 6.2, the behavior index (β_i) for each of the well-behaving nodes would be $\beta_i = 1 - \lambda(N-k) = 1 - \frac{N-k}{2N-3} = \frac{N+k-3}{2N-3}$. Also, assume that each of the well-behaving nodes made an accusation against each of the $N-k$ malicious nodes; then based on Equation (6.2) in Section 6.2, $\omega_i = \frac{N+k-3}{2N-3} - (\frac{N-k-1}{2N-3}) = \frac{2k-2}{2N-3}$.

By Equation (6.3), the certificate of any misbehaving node j, is revoked if $R_j = k\frac{2k-2}{2N-3} \geq R_T$. Which implies that $2k^2 - 2k - R_T(2N-3) \geq 0$; that is, $k \geq \frac{2+\sqrt{4+8R_T(2N-3)}}{4}$. □

Example : Consider a MANET with 100 nodes, if $R_T = \frac{100}{2}$ then $k \geq 70.68$; if $R_T = \frac{100}{3}$, $k \geq 57.80$ or if $R_T = \frac{100}{4}$, $k \geq 50.13$. These values of k are for the worst case scenario where the malicious nodes choose to accuse all the well-behaving nodes of misbehavior and in so doing, increase the probability of they been more speedily identified as being malicious. If anything other than the worst case is assumed, the values for k would be smaller, that is, a smaller number of well-behaving nodes would be necessary to guarantee that identified malicious nodes are incapable of preventing the revocation of their certificates.

6.3.2 Computation and communication overhead

Every network security scheme has some associated computation and communication overhead. Our certificate revocation scheme mainly uses message integrity code (MIC)—which can be computed very efficiently—for message origin and integrity checks. Digital signatures are utilized only for authenticating profile table messages and hash chain y_n values when new hash chains are computed. Profile table messages are sent very infrequently: only when a new node enters the MANET; and if the hash chains are made long enough, one or two hash chains per node, that is, one or two y_n value(s) per network session should suffice. Therefore the signing and verification of signatures for profile table messages and y_n hash chain values should have limited effect on the performance of the certificate revocation scheme owing to the infrequency with which these operations occur.

The communication overhead depends on the total number of nodes N in the MANET, the number of misbehaving or malicious nodes, and the value of the configurable time interval T mentioned in Section 6.2.1. The data the protocol transmit are the profile table and the certificate of each node whenever a new node enters the network. Additionally, each node sends a 64-bit membership confirmation message, plus the 128 or 160-bit MIC every T minutes, which accounts for bandwidth utilization of approximately $3.4 * N * T$ bits/second. The bandwidth utilizes for the broadcast of accusation information depends on the number of malicious or misbehaving nodes in the network.

6.3.3 Communication complexity

In this section we derive the communication complexity of our certificate revocation protocol. We are interested in knowing how many accusation information messages are required to revoke a certificate. The computation is simple in the case where there is only one adversarial node, say node j. If a well-behaving node i is accused by the adversary, then $A_i = 1$, $\alpha_i = 0$, $\beta_i = 1 - \lambda$ and $\omega_i = 1 - \lambda$ (recall from Section 6.2 that A_i is the total number of accusations made against node

i, α_i is the number of accusations (minus 1) made by node i, β_i is the behavior index and ω_i is the weight of node i accusation). Similarly, based on Equation (6.3) in Section 6.2, $R_j = \sum_{i \neq j} \omega_i$, since $\sigma_{ij} = 1$. If a malicious node j makes n accusations against the nodes in the set $\mathcal{N}$, then we need N' nodes to accuse node j of misbehavior. Therefore

$$R_j = \sum_{i \in \mathcal{N}} \omega_i + \sum_{i \notin \mathcal{N}} \omega_i = a(1 - \lambda) + (N' - 1 - a) = N' - 1 - \lambda a \geq R_T$$

Hence, node j certificate is revoked if $N' \geq 1 + \lambda a + R_T$. In the general case, there is a set $\mathcal{A}$ of $K \leq N/2$ adversarial nodes. Let α_{ij} denotes the number of accusations (minus one) made by well-behaving node i after accusing an adversarial node j. As is the case for the single adversarial node (outlined above), to revoke the certificate of one adversarial node, we need N' such that:

$$R_j = \sum_{i \notin \mathcal{A}, i \leq N'} (1 - \lambda A_i - \lambda \alpha_{ij}) = N' - K - \lambda \sum_{i \leq N'} A_i - \sum_{i \notin \mathcal{A}, i \leq N'} \alpha_{ij} \geq R_T$$

The above is obtained by combining Equations (6.1), (6.2) and (6.3) in Section 6.2. The minimum N' required is:

$$N' \;=\; K + \lambda \sum_{i \leq N'} A_i + \sum_{i \notin \mathcal{A}, i \leq N'} \alpha_{ij} + R_T \tag{6.4}$$

Since the well-behaving nodes make accusations in random order, we compute the expected value of N'. There are K adversarial nodes such that $K < N/2$, therefore:

$$\sum_{i \leq N'} A_i \;\leq\; (N - K)K \leq \frac{N}{2}(N - 1) \tag{6.5}$$

Since we do not know the total number of accusations that a well-behaving node i will make, we approximate the expected value of α_{ij} to be $\frac{K}{2}$, which is half of the maximum number of accusations it can make, that is:

$$E\left[\sum_{i \notin \mathcal{A}, i \leq N'} \alpha_{ij}\right] \approx E[N'] \cdot \frac{K}{2} \tag{6.6}$$

Solving for expected value of N' by substituting Equations (6.5) and (6.6) into (6.4), we obtain:

$$\begin{aligned} E[N'] &\leq \frac{1}{1-\lambda K/2}\left[K + \lambda\frac{N}{2}(N-1) + R_T\right] \\ &\leq \frac{1}{1-\frac{1}{4(2-3/N)}}\left[\frac{N}{2}\left(1+\frac{1-1/N}{2-3/N}\right) + R_T\right] \\ &\approx \text{linear in } N \end{aligned}$$

where $\lambda = 1/(2N-3)$.

This implies that a linear number of accusation information broadcasts (which cost order N^2 messages) are sufficient to revoke the certificate of an adversarial node.

6.4 Simulation setup and results

We simulated the protocol using NS2 network simulator. The aim of the simulation is to determine average case performances of the scheme with regards to its effectiveness in revoking the certificates of identified malicious nodes; and in particular to ascertain the average number of accusations necessary to cause the revocation of certificates for various combinations of number of well-behaving nodes versus number of malicious nodes. The process of identifying malicious nodes is beyond the scope of this thesis; however, techniques such as those employed in [50, 31] can be utilized. For the purpose of the simulation, we assumed that if a malicious node m_i made less than $\frac{N}{4}$ accusations (where N is the total number of nodes in the network), there is a probability of 0.50 that a given well-behaving node n_j will identify m_i as being malicious when n_j receives an accusation message from m_i; whereas if m_i made more than $\frac{N}{4}$ accusations, there is a probability of 0.75 that n_j will identify m_i as being malicious when n_j receives m_i accusations.

The simulation attempts to balance the following desires of the malicious nodes: (a) Prevent the revocation of their certificates by reducing the weight of the accusations of well-behaving nodes through malicious accusations. (b) Act in collusion with other malicious nodes and cause the revocation of well-behaving nodes' certificates by maliciously accusing targeted nodes. These two eventualities require different approaches. The former is best achieved if each of the malicious nodes launches accusation against all of the well-behaving nodes; whereas the latter needs conservatism regarding the number of accusations a node makes (see Equation (6.1) and (6.2) in Section 6.2). We used the following simple heuristic for achieving a balance between these conflicting requirements: When a malicious node m_i receives a message from a well-behaving node n_j, if m_i has not previously accused n_j of misbehavior and m_i made less than $\frac{N}{4}$ accusations and the output from a random number generator (which outputs 0 or 1) is 0, then m_i broadcasts an accusation against n_j. In other words, there is a 0.50 probability that a malicious node m_i will accuse a well-behaving node n_j of misbehavior whenever m_i receives a message from n_j; provided that m_i has not previously accused n_j, and m_i made less than $\frac{N}{4}$ accusations. If m_i however made more than $\frac{N}{4}$ accusations and all else being equal, then the probability that m_i launches an accusation against n_j—when it receives a message from the latter—decreases to 0.25. On the other hand, when a well-behaving node n_i receives an accusation message from a malicious node m_j, if n_i has not previously accused m_j, and m_j made less than $\frac{N}{4}$ accusations, there is a probability of 0.50 that n_i broadcasts an accusation against m_j. Whereas the probability increases to 0.75 if m_j made more than $\frac{N}{4}$ accusations. Regarding the collusion aspect of the malicious nodes, when a malicious node m_i receives an accusation against a well-behaving node n_j from another malicious node, if m_i has not previously accused n_j of misbehavior, m_i immediately launches an accusation against n_j. In so doing, malicious nodes can effectively target non-malicious nodes in attempt to blackmail them and cause the revocation of their certificates.

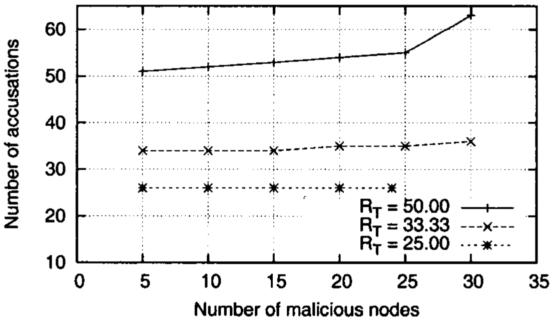

Figure 6–3: Simulation results for 100 nodes

We simulated a MANET environment running destination sequence distance vector (DSDV) as the routing protocol, and examined the performance of our

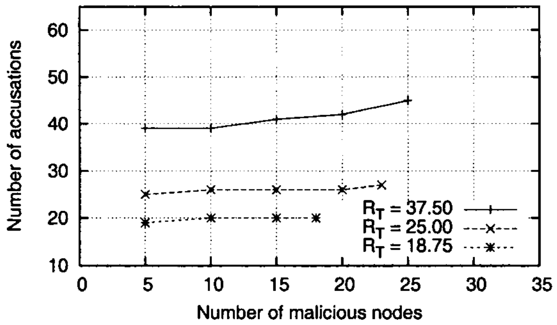

Figure 6–4: Simulation results for 75 nodes

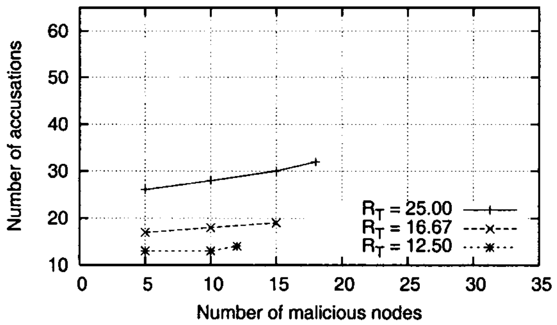

Figure 6–5: Simulation results for 50 nodes

certificate revocation scheme when the number of malicious nodes varies from 5 to x, where x is less than the revocation quotient threshold (R_T), for R_T values of $\frac{N}{2}$, $\frac{N}{3}$ and $\frac{N}{4}$ when N (number of nodes) equals to 100, 75 and 50.

As expected from intuition, the simulation results indicate that generally, as the number of malicious nodes increases, a slightly larger number of accusations are required to cause the revocation of a malicious node's certificate. The exception being when R_T equals $\frac{N}{4}$ for larger values of N, as is the case for N equals 100 (Fig. 6–3) and N equals 75 (Fig. 6–4). Fig. 6–3 for example, shows that when R_T equals 25.00, only 26 accusations are necessary to cause the revocation of a malicious node's certificate, irrespective of the number of malicious nodes (M) present, as M varies from 5 to 24. The lack of influence of the malicious nodes in this regard can be attributed to the following: with $R_T = \frac{N}{4}$ and the number of malicious nodes being less than R_T, the ratio of well-behaving nodes to malicious nodes is higher as the value of N increases. For example, when N equals to 100, the ratio of well-behaving nodes to malicious nodes (M) ranges from 19 to 3 when M varies from 5 to R_T; whereas when N equals 50, this ratio ranges from 9 to 3 as M varies from 5 to R_T. For lower R_T values, higher ratio of well-behaving to malicious nodes has the effect of diluting the influence of the malicious nodes, since smaller percentages of the available well-behaving nodes are sufficient to cause the revocation of a malicious node's certificate.

Another deviation in the results from what is expected from intuition is the higher than average increase in the number of accusations required to revoke a certificate when the number of malicious nodes increases from 25 to 30 or from 20 to 25 for N equals 100 or 75 respectively, when R_T equals $\frac{N}{2}$. This can be attributed to the accumulative effect of the increasing number of malicious nodes. Higher R_T values necessitate larger number of accusations to cause the revocation of a certificate. The malicious nodes therefore have more opportunity to accuse well-behaving nodes before their certificates are revoked. Consequently for higher R_T values, as the number of malicious nodes increases, their effect becomes more pronounced.

In summary, the simulation results indicate that the number of accusations in excess of R_T that is necessary to cause the revocation of a malicious node's certificate depends on the size of the network (N) and the value of R_T. For lower R_T values, that is, for $R_T \leq \frac{N}{3}$, the effect of increasing number of malicious nodes is less pronounced as the size of N increases. However when R_T is greater than $\frac{N}{3}$, the effect of increasing number of malicious nodes is more pronounced for larger networks. In this regard, the simulation results show that when $R_T \leq \frac{N}{3}$, $\lceil R_T \rceil + 4$ accusations are sufficient to cause the revocation of a malicious node's certificate irrespective of the number of malicious nodes (k) in the network, provided that $k < R_T$; whereas, when $R_T > \frac{N}{3}$, as many as $\lceil R_T \rceil + 10$ accusations may be required to cause the revocation of a malicious node's certificate. In light of these results, it may be advantageous for R_T to be less than or equal to $\frac{N}{3}$, provided that the number of malicious nodes (k) in the network is expected to be less than this value. If the latter cannot be guaranteed, then R_T should be increased such that it is always greater than k.

CHAPTER 7
A secure MANET routing protocol with resilience against byzantine behaviors of malicious or selfish agents

In this chapter, we present a secure on-demand[1] multi-path source routing protocol, called RSR (Robust Source Routing).

7.1 Overview of RSR

RSR has two phases: route discovery and route utilization and maintenance phases. We give an overview of each phase below.

Route discovery

In the route discovery phase, a source node S broadcasts a route request indicating that it needs to find a path from S to a destination node D. In the route request, S stipulates that the path it seeks must not contain any node which is listed in its tabu list, or any link that appears in its exclusion links list. We provide a rationale for the tabu list and exclusion links list in Section 7.3. Additionally, the path must not contain any node which is found in the tabu list of an element in the path. Each node through which the route request traverses is required to append its identifier and its tabu list to the appropriate field of

[1] On-demand protocols have been shown to perform better and have significantly lower associated overhead than proactive protocols [17, 60, 81].

the route request, and signs the packet. Therefore, the information regarding the identity of the nodes that should be excluded from the path is easily ascertained. When the route request packets arrive at the destination node D, D selects three valid paths, copy each path to a route reply packet, signs the packets and unicasts them to S using the respective reverse paths. S proceeds with the utilization and maintenance phase when it receives the route reply packets.

Route utilization and maintenance

The source node S selects one of the routing path it acquired during the routing discovery phase, and sends the data traffic. The destination node D is required to send a signed acknowledgment for each data frame it receives. If S does not get an acknowledgment from D for a data frame after a given number of retries; and it does not receive a link-layer error message indicating that the destination D is unreachable, S assumes that there are selfish or malicious nodes on the path and proceeds as follows: S constructs and sends a forerunner packet to inform the nodes on the path that they should expect a specified amount of data from the source of the packet within a given time. When the forerunner packet reaches the destination, it sends an acknowledgment to S. If S does not receive an acknowledgment for the forerunner packet, it proceeds as outlined in Section 7.3.2, under the heading "No ACK for a FR packet returns from D." Otherwise, S commences the data traffic flow to D. If there are selfish or malicious agents in the path and they choose to drop the data packet or acknowledgment from D, such eventuality is dealt with as outlined in Section 7.3.2, under the heading "S commenced data flow to D but the traffic is being dropped."

7.2 Problem definition and model

In this section we outline the network and security assumptions we utilized in the design of RSR. We also present a more precise description of the problem our protocol addresses.

7.2.1 Network assumptions

RSR utilizes the following assumptions regarding the targeted MANETs:

- Each node has a unique identifier (IP address, MAC address or certificate serial number).
- Each node has a valid certificate and the public keys of the CAs which issued the certificates of the other network peers.
- The wireless communication links between the nodes are symmetric; that is, if node n_i is in the transmission range of node n_j, then n_j is also in the transmission range of n_i. This is typically the case with most 802.11 [54] compliant network interfaces.
- The link-layer of the MANET nodes provide transmission error detection service. This is a common feature of most 802.11 wireless interfaces.
- Any given intermediate node on a path from a source to a destination may be malicious and therefore cannot be fully trusted. The source node only trusts a destination node, and visa versa, a destination node only trusts a source node.

7.2.2 Threat model

In this work, we do not assume the existence of security association between any pair of nodes. Some previous works, for example [91, 47] rely on the assumption that protocols such as the well known Diffie-Hellman key exchange protocol [28] can be used to establish secret shared keys on communicating peers. However, in an adversarial environment, malicious entities can easily disrupt these protocols—and prevent nodes from establishing shared keys with other nodes—by simply dropping the key exchange protocol messages, rather than forwarding them. Our threat model does not place any particular limitations on adversarial entities. Adversarial entities can intercept, modify or fabricate packets; create routing loops; selectively drop packets; artificially delay packets; or attempt denial of service attacks by injecting packets in the network with the goal of consuming network resources. Malicious entities can also collude with other malicious entities in attempts to hide their adversarial behaviors. The goal of our protocol is to detect selfish or adversarial activities and mitigates against them.

One particular type of attacks our protocol cannot prevent is wormhole exploits [48]. In wormhole attacks, an attacker receives packets at one point in a network, tunnel them to another point in the network and replays them into the network from that point. Colluding adversaries can use this attack, for example to forward route request packets in attempt to increase the likelihood of adversarial entities controlling routing paths. If a wormhole exhibits adversarial activities, our protocol mitigates against these exploits by treating the wormhole as a single link and make efforts to avoid utilizing it.

7.2.3 Problem definition

Our goal in this work is to provide a robust on-demand secure routing protocol which operates under the assumption listed in Section 7.2.1 and mitigates against any of the possible adversarial activities outlined in the threat model in Section 7.2.2. The explicit aim of the protocol is not to eliminate adversarial activities—since it is virtually impossible to prevent some of these activities—rather, the objective is to discourage selfishness and lessen the effects of the adversarial activities indicated in the threat model.

7.3 Details of RSR

The protocol requires each node to keep a tabu list containing a list of nodes which the owner of the list deems malicious or untrustworthy. The owner of the list will silently drop route request packets originated from any node that is in its tabu list. It is therefore highly likely that the owner of a tabu list will be listed in the tabu lists of the nodes in its tabu list. Hence, it is within a node's best interest to add a node to its tabu list only if it has a high degree of certainty that the given node is malicious or untrustworthy.

As previously indicated, the routing scheme consists of two phases: route discovery and route utilization and maintenance phases. All unicast routing packets transmitted in each phase of the protocol have a common source route header with the following fields:

- *Source address*: The identifier of the node which constructed the packet.
- *Destination address*: The identifier of the destination node.

- *Source route*: The routing path the packet must traverse in transit from the source to the destination.

7.3.1 Route discovery

When a node n_i has data to transmit to a destination which it does not know of a path to, n_i generates a route request (RREQ) packet containing the following information:

- *Request id*: A unique, random nonce, which together with the source address serves as the identifier of a RREQ packet.
- *Exclusion links*: A list of zero or more link(s) which must not be included on a path.
- *Route record*: The list of nodes the RREQ traverses, along with their tabu lists and accompanying signatures.

It should be noted that exclusion links and the tabu lists are separate entities which serve different purposes, namely: when a node n_j is listed in node n_i's tabu list, n_i will silently drop RREQ packets originated from n_j. If n_i is currently on any of n_j's routing paths, it will continue to forward data traffic along the given path(s); however, n_i will not appear on any new path for n_j since it will not forward any other RREQ packets from n_j. On the other hand, if n_j appears on a link in n_i's exclusion links, n_i will still continue to forward RREQ packets originated from n_j; since n_i does not know whether it is n_j or n_k (the other node in the problematic link) that is the selfish or adversarial node.

After generating the RREQ, n_i signs the RREQ and broadcasts the packet to its neighbors. When a node n_j receives a set of RREQ packet—it has not previously seen—with the same ⟨source address, request id⟩ identifier, it selects one at random[2] then checks if any of the following holds:

- The source of the RREQ is listed in n_j's tabu list.
- n_j appears in a tabu lists in the route record field.
- There is an exclusion link between n_j and a neighbor which appears in the route record field.

If any of the above holds, n_j discards the packet and records that it has seen a RREQ with the given ⟨source address, request id⟩ identifier. Otherwise, n_j verifies the initiator's signature[3] ; if the verification fails and n_j's link-layer does not report a transmission error, n_j adds the neighbor it received the RREQ packet from to its tabu list and discards the RREQ. The reason being, n_j's neighbor either modified or fabricated the packet, or it did not verify the source's signature before forwarding the RREQ; that is, n_j's neighbor is either malicious or it is not complying with the protocol. If the signature verification succeeds, n_j appends its identifier and its tabu list to the route record field, signs the entire route record field, makes a record indicating that it has seen a RREQ packets with the given ⟨source address, request id⟩ identifier, and broadcasts the packet to its neighbors.

[2] Selecting an RREQ packet at random rather than choosing the first one that arrives provides protection against rushing attack [49].

[3] Source authentication is utilized to extenuate the effect of denial of service attacks on the network. We discuss the pros and cons of this approach in Section 7.4.

RREQ packets continue to traverse the network in the manner described above until one or more reach the destination node D. On receiving a list of RREQ packets with the same ⟨source address, request id⟩ identifier, node D is expected to select three of the RREQ packets such that the path in their respective route record field has the least number of hops, and no element in the path appears in any of the other path elements' tabu lists and no link is listed in source's exclusion links. Next, D is required to verify the signatures in the route record fields of each of the selected RREQ packets. If the signatures of a selected RREQ packet are all valid, D constructs a route reply (RREP) packet for the given RREQ, signs it and unicasts it—using the reverse path in the RREQ route record field—to the source of the RREQ. If any of the signature verification for a selected RREQ packet fails, the RREQ in question is discarded and another selected using the criteria outline above. The source node S is expected to send a signed acknowledgment for each RREP it receives. If D does not get an acknowledgment from S for a RREP packet after a given number of retries; if there are other RREQ packets remaining, D selects another, processes it as outlined above and sends the resulted RREP packet to S.

In addition to the common source route header, a RREP packet contains the following information:

- *Request id*: Request id of the corresponding RREQ packet.
- *Path*: The identifiers of the nodes in the routing path, in the order indicated in the route record field of the corresponding RREQ.

When the source of the RREQ receives the RREP packets, it proceeds with the route utilization and maintenance as indicated below.

7.3.2 Route utilization and maintenance

On receiving the RREP packets, the source node S stores the paths, selects one which has the least number of hops, and proceeds to send the data traffic. The destination node (D) is required to send a signed acknowledgment (ACK) for each data frame it received. If S does not received a valid ACK for any given data after a certain number of retries, nor does S received a link-layer error message from any of the intermediate nodes; S assumes that there is/are selfish or malicious node(s) on the given path, and proceeds with the fault detection and isolation phase below.

Fault detection and isolation

When there is evidence of misbehaving node(s) on a given path, the protocol utilizes a forerunner (FR) packet to inform the nodes on the path that they should expect a certain data flow rate from S to a specified destination. The intention being that if any of the path elements do not receive the specified data traffic within a configurable time period after receiving a FR packet from S, it will send a negative acknowledgment, informing S that it did not receive the expected data flow. Data flow rate can be obtained from IEEE 802.11 MAC (Medium Access Control) protocol operating in the DCF (Distributed Coordination Function) mode, using mechanisms outlined in [22, 107, 66].

A *forerunner (FR)* packet has the following fields:

- *FR id*: A unique, random nonce, which together with the source address (ascertained from the source route header) serves as the identifier of a FR packet.
- *Expected data rate*: data flow rate which should follow the FR packet.
- *ACK indicator*: This is a 1-bit flag which is set if the intermediate nodes are required to send a signed ACK back to the source of the FR packet.

To avoid unnecessary network traffic, the ACK indicator flag is set to 0 when a FR packet is constructed. The packet is then signed and sent to D using the selected path. When an intermediate node on the path from S to D receives the FR packet, it is expected to verify the signature, if it is valid, it should note the time it received the FR packet then forward the packet to the next hop on the path. When D receives a valid FR packet, it sends a signed ACK back to the source. On receiving the ACK from D, S commences the traffic flow to D.

Selfish or malicious nodes may choose not to forward a FR packet, and they also may not forward data traffic after S commences the traffic flow to D. The protocol deals with these eventualities as indicated below:

No ACK for a FR packet returns from D

If S does not receive an ACK for a FR packet from D, nor does S received a link-layer error message from any of the intermediate nodes indicating that the destination D is unreachable; S assumes that a misbehaving node on the given path has dropped the FR packet or the ACK from D, and proceeds as follows: if the length of the path from S to D is exactly 3, S adds the link between D and the intermediate node to its exclusion links, discards the path, selects another path to D—if one is available—and repeats the route utilization process indicated above. If there are no more precomputed path to D, S constructs, signs and broadcasts another RREQ packet with the exclusion link field containing all the problematic link(s) it has recorded. If the path length from S to D is greater than 3, S constructs another FR packet, sets the ACK indicator flag to 1, signs the packet and sends it to the first hop on the path to D. When a node n_i receives a FR packet with the ACK indicator flag set to 1, n_i is expected to broadcast—via limited flooding—a signed ACK back to S. In the limited flooding broadcast, the time-to-live (TTL) field of the IP header is set to d where d is the number of hops from the node in question to S. If S does not receive a valid ACK from each of the nodes in the path, then the link between the first node n_i—on the given path, from which S does not receive a valid ACK—and n_i's upstream path neighbor is added to S exclusion links. For example, in Fig. 7–1, if S receives ACKs for the FR packet from $n1$ and $n2$ but not from $n3$, S would add the link between $n2$ and $n3$ to its exclusion links, selects another path to D or sends out a route request as outlined above. A path with a problematic link can be pruned by removing the sub-path commencing with the downstream node of the problematic link. For example, in Fig. 7–1, $n3$ and D would be removed from

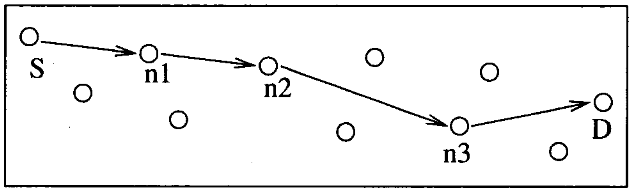

Figure 7–1: A routing path example

the path; resulting in a sub-path of length 3 from S to $n2$. The resulted sub-path after the pruning operation is stored if its length is greater than or equal to 3, or discarded otherwise.

S commenced data flow to D but the traffic is being dropped

As indicated above, when a node n_i receives a FR packet, it records the time it received the packet. If a configurable time period (which depends on the network latency and available bandwidth) passed and n_i does not receive the expected data flow from S to D, n_i is required to send—via limited flooding—a signed negative ACK to S, indicating that n_i has not received the data flow it expects from S. A negative ACK is similar to an ACK packet, except that it informs the intended recipient S that the source of the negative ACK did not receive the data traffic it expected from S. When S receives a valid negative ACK from a node n_i, and it is confirmed by other negative ACKs from downstream nodes on the path to D, S records the link between n_i and n_i's upstream path neighbor as being problematic; S then prunes the given path and repeat the process of selecting or discovering another path, as outlined above.

Rather than dropping data traffic, malicious nodes may choose to tamper with the data. The protocol deals with this eventuality by requiring intermediate nodes to verify the source's signature on packets they received, before forwarding them. If the signature verification fails for node n_i, and n_i link-layer does not report a transmission error, n_i is required to add the neighboring node it received the packet from to its tabu list, and sends—via limited flooding—a negative ACK to S, informing it that the packet has been modified. On receiving the negative acknowledgment from n_i, S is expected to append the link involving n_i and its upstream neighbor to its exclusion links, and prunes the path.

7.4 Discussion

In this section, we elaborate on relevant design choices of our protocol. We commence with our choice of using digital signatures for integrity checks and source authentication.

7.4.1 Choice of cryptographic tools

Most network security schemes utilize message authentication codes, rather than digital signatures, for integrity checks. This is so due to the fact that message authentication codes can be computed much more efficiently than digital signature computations. The drawback for the use of message authentication codes, as is the case for other symmetric-key cryptographic tools, is that it requires shared keys to be established among the communicating peers. As alluded to in Section 7.2.3, our protocol was specifically designed for adversarial MANET environments which contain, or is likely to contain persistent malicious or selfish entities which seek to disrupt the network by perpetrating the adversarial activities outlined in the threat model in Section 7.2.2. We argue that it may not be feasible to establish shared keys among communicating peers, using key exchange protocols, since adversarial entities can easily thwart these protocols by dropping the protocol messages, rather than forwarding them. A node S can generate a symmetric key, signs it, encrypts it with the public key of the intended recipient and sends it via broadcast to the destination D. This will likely allow a shared key to be established between S and D; however the cost in throughput reduction, due to the extra broadcast messages, may not justify this approach. Alternatively, shared secret keys can be distributed to the network nodes using appropriate out-of-band means; again, this approach is not feasible considering the likelihood of shared keys being compromised if they are not refreshed frequently.

Aside from the problem of establishing shared secret keys among communicating peers in highly adversarial environments, message authentication codes may not be as effective in identifying certain malicious activities. For example, a malicious entity, on a routing path from S to D, which seeks to disrupt the traffic flow on this path, can choose to illicitly modify packets and forward them rather than mere dropping the packets. The end result of these activities is similar to packet dropping since the destination will discard the packets when it ascertains that they have been illicitly modified. Digital signature can be used to identify malicious entity which modified the packet, or identify the colluding malicious entity which forwarded the modified packet; but message authentication code is lacking is this regards, since typically only the source and the destination of a traffic flow know the secret key for computing the message authentication codes. We leveraged the aforementioned feature of digital signature in the design of RSR to help to detect and isolate adversarial entities. RSR source authentication operations serves two main purposes:

1. Consider for example the scenario shown in Fig. 7–1. If $n3$ (a well-behaving node) receives a packet from $n2$ to forward to D, if the signature verification for the packet fails and $n3$ link-layer does not report a transmission error, $n3$

will add $n2$ to its tabu list. The reason being, either $n2$ modified the packet or it did not verify the signature on the packet; that is, $n2$ is either malicious or it is not complying with the protocol. In addition to adding $n2$ to its tabu list, $n3$ will discard the packet and send a negative acknowledgment to S informing it that the packet it received has been illegitimately modified. If this info from $n3$ is supported by the fact that S does not receive an acknowledgment from D for the given data frame, S will add the link between $n2$ and $n3$ to its exclusion links and consequently commences the process of isolating $n2$.

2. Source authentication can also be use to attenuate certain denial of service exploits. Malicious nodes may attempt to flood the network with fabricated packets in attempts to consume network resources. RSR source authentication operations are partly aimed at reducing the effect of these types of attacks by stipulating that nodes discard unauthenticated packets. It should be noted that adversarial entities can overwhelm individual nodes in their one-hop neighborhood by sending them large number of fabricated packets. However, the fact that the unauthenticated packets will be discarded, the resource consumption exploit will be limited to the one-hop neighborhood of the adversarial entities.

In light of the above possibilities, it is our view that the benefits of using digital signature for source authentication outweighs the associated cost. Digital signature schemes such as RSA [103] allow trade-off between signing and verification operations. If the public exponent of the crypto system is small, verification can be several times faster then signing operations. Example, for a 1024-bit RSA key, if the public exponent (e) is 3, verification operations can be over 700 times faster than signing operations [119]. Verification of signatures can therefore be done fairly efficiently; most of the digital signature operations in RSR are verification activities.

An alternative approach to utilizing cryptographic tools for the operations outlined in item 1 above, is to have the nodes' network interfaces operate in promiscuous mode and stipulate that the nodes monitor the traffic that flows in and out of each of their neighbors, and report all discrepancies. This operation however is inefficient and is subjected to the short comings outlined in Section 5.1.3 for Marti *et al* scheme [82].

7.4.2 Tabu list and exclusion links

RSR utilizes tabu lists and exclusion links to record problematic nodes and links, respectively. The consequences of being listed in a node's tabu list is more severe for the following reasons: a node will silently discard route requests from nodes which are listed in its tabu list. Therefore, if a node is listed in the tabu lists of several nodes, it will likely have much difficulties communicating with other nodes which are not in its transmission range. On the other hand, a node's exclusion links list is used solely to exclude problematic links from its routing paths. This design choice is motivated by the fact that a node n_i does not know for sure which element of a problematic link is selfish or adversarial; and n_i wants to avoid the possible of wrongfully isolating well-behaving nodes. Malicious nodes may add well-behaving nodes to their tabu list with the intention of disrupting route discovery processes; however, this eventuality would actually have positive effects on the network, since this reduces the possibility of the given malicious nodes being on routing paths. Similarly, adversarial entities will not achieve any benefit from adding functional links to their exclusion links lists.

7.4.3 Forerunner packets mechanism

Our Forerunner (FR) packet mechanism requires MANET node to be able to determine the flow rate of incoming traffic. As outlined in Section 7.3.2, data flow rate can be obtained from IEEE 802.11 MAC protocol quite efficiently using techniques presented in [22, 107, 66]. The distinguishing feature of our forerunner (FR) packets mechanism compare to other MANET fault detection techniques—such as probing—is the following: FR packets inform the nodes on a path from a source node S to a destination node D that S intends to send a certain amount of data within a given time period; therefore the nodes should expect the specified data traffic flow rate from S for the time period indicated. If the nodes on the path from S to D do not receive the specified data traffic flow rate within the specified time period, they are required to send negative acknowledgments to S informing S that they did not receive the expected data flow. This mechanism forces selfish or malicious entities on routing paths to cooperate and forward the specified data traffic a FR packet announced would follow, or risk being identified as problematic if they choose not to forward the data traffic. The selfish or malicious entities can resume adversarial activities after forwarding the specified data traffic a FR packet announced. However, the end result is that FR packets can force uncooperative entities to forward specified amount of data; or conversely, help to identify links which contains uncooperative nodes. This can be contrasted with schemes such as [6, 63, 65] which utilize probing techniques, in that the probing mechanisms will succeed in enforcing cooperation only if the probing packets are completely indistinguishable from other data packets; which in reality is very difficult to achieve. There are no needs for FR packets to be indistinguishable from other packets since their purpose is to announce intended traffic flows. Adversarial entities can choose to drop FR packets; however as outlined Section 7.3.2 and 7.5, the protocol operations provides means for identifying these adversarial activities.

7.5 Analysis

In this section we give specific examples of malicious behaviors and show how RSR mitigates against these possible exploits.

7.5.1 A single malicious node on a routing path

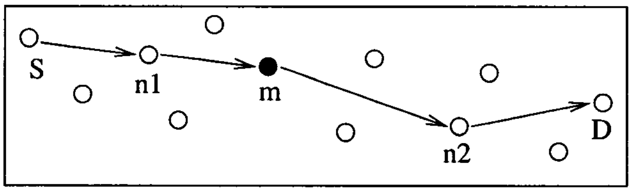

Figure 7–2: One malicious node on a routing path

Consider the following with respect to the routing path depicted in Fig. 7–2:

1. If m drops a data packet sent from S to D, S would not receive an ACK from D for the given packet. Consequently, S sends a FR packet along the path to D. If m drops this packet, no acknowledgment will return from D

for the FR packet. S will then send a FR packet with the ACK indicator bit set to 1, along the same path to D. Each node along the path that receives the FR packet (with the ACK indicator bit set to 1) is required to send—via limited flooding—a sign ACK to S. If m drops this packet, S will not receive an ACK from $n2$. Therefore, S will classify the link between m and $n2$ as problematic and adds it to its exclusion links. The next RREQ packet S sends out will contain information about the faulty link between m and $n2$. When $n2$ receives this info, if there were at least $N - 1$ (see Section 7.4.2 for info related to N) other RREQ packets from different sources which listed this link as problematic, $n2$ will add m to its tabu list; thus initiating the process of isolating m.

2. If m acknowledges and forwards the FR packets with the ACK indicator bit set to 1, but succeeded—with the help of other malicious nodes outside the given path—to filter out the ACKs sent by $n2$ and D to S, S will not get an ACK from $n2$. Therefore, S will add the link between m and $n2$ to its exclusion links.
3. If S receives an ACK for the FR packet (with the ACK indicator bit set to 1) from each of the path element, S will start sending the specified data traffic to D. If m drops a data frame, $n2$ and D would not receive the data flow the FR packet specified that they should expect. Consequently, they will send a negative ACK—via limited flooding—to S. When S gets the negative ACKs, S adds the link between m and $n2$ to its exclusion link.

4. If m with the help of other malicious nodes outside the given path, succeeds in filtering out the negative ACKs from $n2$ and D, S will know that the path has a fault, since it does not receive an ACK from D for the data frame m dropped. Consequently, S will discard the given path. The same holds if m forwards all the data frames from S to D but drops an ACK D sends to S.

7.5.2 Colluding malicious nodes adjacent to each other

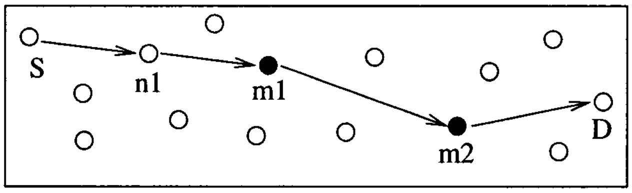

Figure 7–3: Adjacent colluding malicious nodes on a routing path

Consider the path shown in Fig. 7–3 with the colluding malicious node $m1$ and $m2$. If $m1$ or $m2$ drops packets they are required to forward, It is trivial to show that the same arguments outlined in 1), 2), 3) and 4) above hold.

7.5.3 Colluding malicious nodes 2 hops away from each other

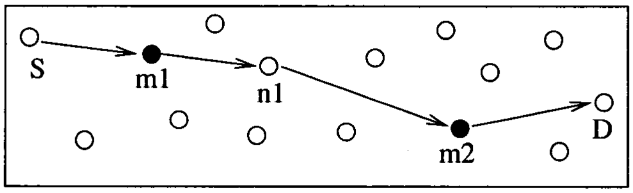

Figure 7–4: Non-adjacent colluding malicious nodes on a routing path

In the path shown if Fig. 7–3, if $m1$ or $m2$ drops packets that were intended to be forwarded to D, it can also be trivially shown that the arguments outlined in

1), 2) and 3) above hold. In this scenario, however, it is unlikely that $m1$ will ever succeed—with the help of other malicious nodes outside the given path—in filter out negative ACKs sent via limited flooding from $n1$ to S, unless all of the nodes that are within S transmission range are malicious. It will therefore be difficult for $m1$ to conceal its malicious behaviors.

7.6 Simulation evaluation

We implemented RSR in NS2 network simulator [1]. For the cryptographic components, we utilized Cryptlib crypto toolkit [43] to generate 1024-bit RSA cryptographic keys for the signing and verification operations. In the simulation implementation, malicious nodes do not comply with the protocol. For example, they do not verify the signatures on the packets they forward, nor do they add nodes to their tabu list or exclusion links, or send negative ACKs. In addition, they selectively drop or modify packets they are asked to forward. The exception being that they do not drop or modify RREQ or RREP packets, since their adversarial effects are more pronounced when they are on as many routing paths as possible. Table 7–1 summaries the simulation parameters.

7.6.1 Performance metrics

We used the following metrics to evaluate the performance of our scheme.

1. **Packet Delivery Ratio**: This is the fraction of data packets generated by CBR (Constant Bit Rate) sources that are delivered to the destinations. This evaluates the ability of RSR to deliver data packets to their destinations in

Table 7–1: Simulation parameters values

Parameter	Value
Space	670 m x 670 m
Number of nodes	50
Mobility model	random waypoint
Speed	20 m/s
Pause time	600 s
Traffic type	CBR
Max number of connections	34
Packet size	512 bytes
Packet generation rate	4 packets/s
Simulation time	170 s

the presence of varying number of malicious agents which selectively drop packets they are required to forward.

2. **Number of data packets delivered**: This metric gives additional insight regarding the effectiveness of the scheme in delivering packets to their destination in the presence of varying number of adversarial entities.
3. **Routing Overhead (bytes)**: This is the total number of bytes of routing control messages generated over the length of the simulation.
4. **Routing Overhead (packets)**: This is the total number of routing control messages generated over the length of the simulation. We normalized the routing overhead by the number of packets sent and the number of packets received, to compensate for the fact that in the simulation implementation adversarial nodes do not sent data packets.
5. **Average end-to-end latency of the data packets**: This is the ratio of the total time it takes all packets to reach their respective destinations and

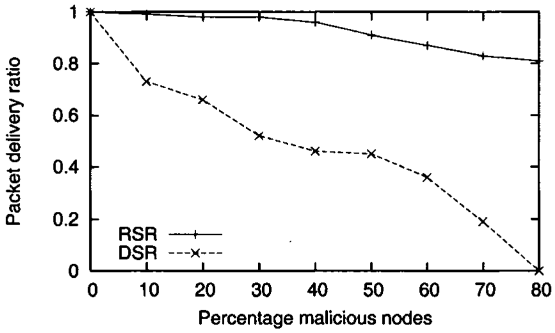

Figure 7–5: Data packet delivery ratio

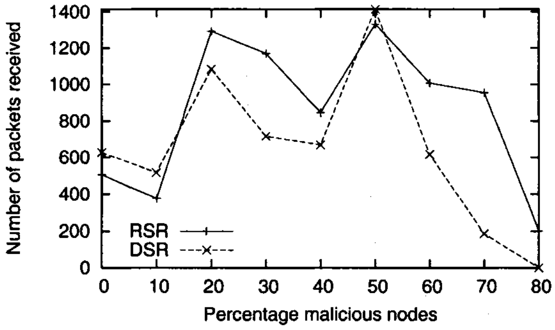

Figure 7–6: Number of packets received over the Length of the simulation

the total number of packets received. This measures the average delays of all packets which were successfully transmitted.

The results of the simulation for RSR is compared to that of DSR [61], which currently is perhaps the most widely used MANET source routing protocol.

7.6.2 Simulation results

The simulation results confirm that RSR is very effective in delivering data packets to their intended destinations even in the presence of large proportion of malicious entities. As indicated in Fig. 7–5, RSR was able to maintain delivery ratio of over 0.8 even when 80 percent of the nodes are malicious. Whereas the

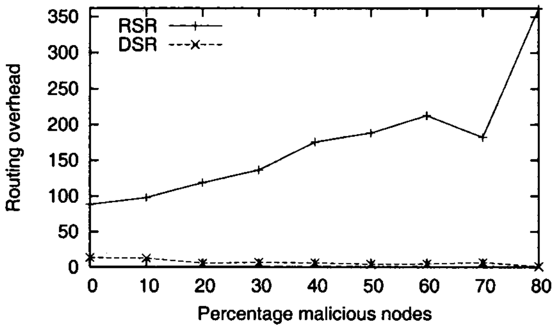

Figure 7–7: Routing overhead (bytes) normalized by number of data packets sent

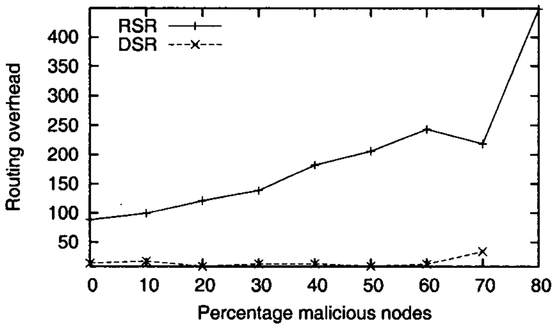

Figure 7–8: Routing overhead (bytes) normalized by number of data packets received

delivery ratio for DSR was 0.2 when 70 percent of the nodes are malicious and 0 when 80 percent of the nodes are malicious.

It should be noted that DSR does not provide any security services, nor does it provide reliable data transfer; whereas RSR provide both of these features. It is therefore expected that the overhead associated with RSR will be significantly higher than DSR. This is the trade-off relating to the overhead of the two protocols. In spite of the higher overhead associated with RSR, Fig. 7–6 indicates that over the length of the simulation, RSR on average, delivers more than twice the number of packets DSR delivers when the percentage of malicious nodes in

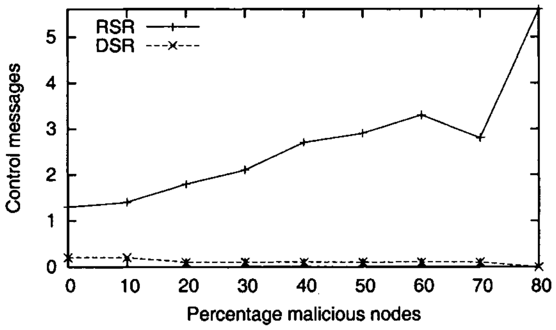

Figure 7–9: Routing overhead (number of packets) normalized by number of data packets sent

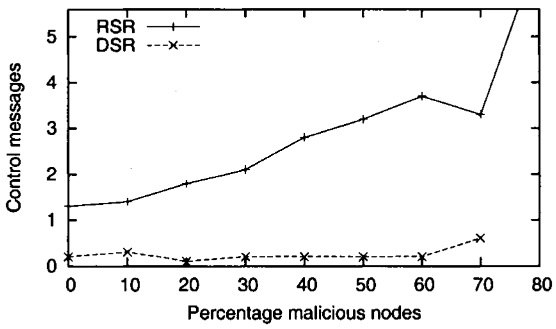

Figure 7–10: Routing overhead (number of packets) normalized by number of data packets received

the network is greater than 10. This confirms—as the plot of delivery ratio (Fig. 7–5) indicates—that in the presence of active malicious entities, RSR allows much greater throughput than DSR.

RSR employs digital signature to provide data origin authentication and integrity checks. In the RSR simulation implementation, each routing packet is signed and the signature appended to the packet; therefore RSR packets are much larger than that of DSR. As expected, the simulation results indicate that the routing overhead for RSR, an average, increases as the percentage of malicious

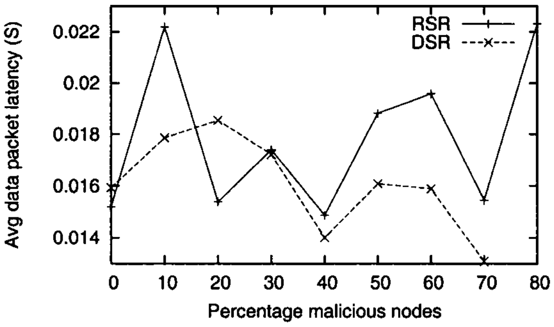

Figure 7–11: Average data packet latency (S)

nodes increases. This is due to the following: as malicious activities increase, more FR packets and consequently ACKs for FR packets, and negative ACKs are sent. Figs 7–7, 7–8, 7–9 and 7–10 indicate that the trends are similar whether the overhead in terms of bytes or packets generated is normalized by number of packets sent or number of packets received.

Fig 7–11 shows that there are no clear trends regarding average data packet latency. The fluctuation in data packet latency is likely related to the number of broadcast packets circulating in the network. The higher the number of broadcast packets in the network, the more contention there will be for the wireless access medium; and consequently, the longer it will take for packets to be delivered to their respective destinations. Average data packet latency is also inversely related to data packet size: larger packets, on average, take longer to reach their destinations. Hence, the higher average packet latency for RSR compared to DSR is expected. However this increase in latency is insignificant compared to the proportionally higher throughput that RSR provides in the presence of increased number of active malicious entities.

One result that is unexpected for RSR is the decrease in overhead when the percentage of malicious nodes increases from 60 to 70 as indicated in Figs 7–7, 7–8, 7–9 and 7–10. This trend is likely related to the CBR traffic pattern during this time interval in the simulation. One possibility is that the CBR data packets sent—during this time interval in the simulation—traverse fewer malicious nodes; consequently there may have been a slight decreased in malicious activities during this time interval.

CHAPTER 8
Conclusion

8.1 Summary

In Chapter two we noted that the security solutions proposed for addressing access control, authentication, integrity and confidentially services for MANETs, utilize one or more of the following cryptographic technologies: symmetric-key cryptography, digital certificates and threshold cryptography. We then gave brief descriptions of the IEEE 802.11 related security standards (WEP, TKIP and CCMP) and other symmetric-key based MANET security schemes. Following this, we presented a brief history of digital certificates, highlighted the distinguishing features of the four main types of digital certificates currently in use, and categorized the existing MANET security proposals which utilize digital certificates. We grouped these proposals into two main categories: schemes which have no preference for digital certificate type, and schemes which require PGP certificate type. We subdivided the group of schemes which have no preference for digital certificate type, into three categories: schemes which do not address certificate revocation, schemes which require access to trusted third party, and schemes which do not require access to trusted third party. Following the presentation related to digital certificates, we described the operations of (k, N) threshold schemes, verifiable secret sharing, proactive secret sharing and identity-based cryptography;

then we reviewed the existing MANET security proposals which utilize threshold cryptography.

In Chapter four we highlighted the pros and cons of the above-mentioned cryptographic technologies. We noted that security schemes which are based solely on symmetric key cryptography are less robust and offer lower degree of security, owing to key management issues associated with shared secret keys. Following this, we discussed the hierarchical and Web-of-trust trust models and argued that the Web-of-trust model is more susceptible to infiltration of malicious agents; therefore schemes which utilize this trust model are not suitable for MANET environments with high security requirements. Next, we highlighted the challenges of utilizing certificates based on hierarchical trust model in MANETs. One of the foremost challenge is the issue certificate revocation in MANETs where there are no on-line access to trusted authorities. We noted that only one of the digital certificate based proposals we reviewed in Chapter two addresses certificate revocation and does not rely on the assumption that access to on-line CAs is available. This scheme however does not provide protection against certificates being wrongfully revoked through malicious accusations. Following the discussion on digital certificates, we analyzed the schemes which employ threshold cryptography and argued that these schemes are not suitable for most MANET environments for two main reasons:

1. The computational overhead associated with threshold cryptography is too prohibitive for low-powered MANET nodes.

2. Threshold cryptographic schemes require unselfish cooperation of the network nodes. This requirement is unrealistic in most non-military network environments.

From the above discussion, we assert that the issue of certificate revocation in MANETs is an important, open research problem.

In Chapter five we analyzed the existing secure MANET routing proposals. We noted that most of these proposals do not mitigate against selfish or malicious entities which selectively drop packets they agreed to foreword. We categorized the proposals which attempt to mitigate against these adversarial activities into three categories: trust-based routing schemes, incentive-based schemes and schemes which employs detection and isolation mechanisms. We argued that trust-based routing schemes are susceptible to adversarial exploits because they either require group secret keys to enforce trust-level requirements, they do not provide protection against malicious accusation attacks, or they can be thwarted by dropping the trust query messages. Next we highlighted the point that the incentive-based schemes either require tamper resistant hardware module or they require on-line access to a centralized entity. Owing to these requirements, the incentive-based schemes are limited in their applications. Regarding the schemes which employs detection and isolation mechanisms, we asserted that these schemes are inadequate for the various reasons outlined in Section 5.1.3. Finally, we concluded from the review and analysis of the existing MANET secure routing proposals, that there are needs for secure routing schemes which adequately mitigate against selfishness and selective packet dropping in MANETs.

8.2 Original contributions

This thesis makes two main original contributions:

1. we present a decentralized certificate revocation scheme for MANET. Our scheme delegates all key management tasks, except the issuing of certificates, to the nodes in a MANET; and it does not require access to on-line CAs. Our certificate revocation protocol has relatively low associated computational overhead owing to the fact that it mainly uses hash chains for the security mechanisms. We present a security analysis in which we outline four possible attacks malicious entities can launch against our certificate revocation protocol and examine how the protocol deals with these adversarial activities. We provide communication complexity analysis which shows that order N^2 accusation info messages are sufficient to cause the revocation of a malicious node's certificate. Finally, we present simulation results indicating the effectiveness of our certificate revocation protocol in revoking the certificates of adversarial entities in such a way that the nodes in the network are cognizant of the certificates revocation information in a timely manner.
2. We present a robust, secure MANET on-demand routing protocol which is capable of delivering packets to their destinations even in the presence of large proportions of active malicious or selfish agents which selectively drop packets they agreed to forward. We named this protocol Robust Source Routing (RSR). RSR introduces the concept of forerunner (FR) packets which inform nodes along a path that they should expect specified data flow

within a given time frame. The path elements can therefore be on the look out for the given data flow, and in the event that they do not receive the traffic flow, they can transmit info to the source informing it that the data flow they expected did not arrive. In so doing, links with active malicious agents can be identified, and the malicious agents be eventually isolated.

To the best of our knowledge, this is the first work that utilizes the concept of forerunner packets to encourage cooperation in MANETs. Finally, we provide simulation results showing that in the presence of increased number of active adversarial nodes, RSR maintains delivery ratios that exceeds those associated with DSR by more than 50 percent. Additionally, the simulation results indicate that on average, RSR provides throughput that is two fold that of DSR, when the percentage of malicious nodes is greater than 10 percent.

To highlight the significance of the contribution of RSR, we reproduce the table (in Table 8–1) containing the summary of the analysis we gave in Chapter 5, along with a brief summary of RSR contributions.

Table 8–1: Summary of routing security analysis

Schemes	Comments
Schemes which do not address packet dropping	SRP [91], SEAD [47], SAODV [127], Ariadne [46], ARAN [105], Binkley *et al* [9] and Venkatraman *et al* [115] schemes do not address packet dropping.
Trust-based schemes	SAR [124] requires shared group keys; therefore it is subjected to the key management issues outlined in Section 4.1.1. Pirzada *et al* and Nekkanti *et al* [98, 88] do not provide protection against packet dropping; SDAR [16] is subjected to the short comings indicated below for Marti *et al* scheme; Li *et al* [78] scheme can be thwarted by dropping the trust query messages.
Incentive-based schemes	Buttyán *et al* [18] requires tamper resistant hardware and Zhong *et al* [130] requires on-line access to a centralized entity; therefore, these schemes are limited in their applications.
Schemes which employ detection and isolation mechanisms	Marti *et al* [82], in the author's own words, has the following weaknesses: "it might not detect a misbehaving node in the presence of 1) ambiguous collisions, 2) receiver collisions, 3) limited transmission power, 4) false misbehavior, 5) collusion, and 6) partial dropping." Buchegger *et al* [104] scheme does not provide protection against false accusations. The probing technique Awerbuch *et al*, Just *et al* and Patwardhan *et al* schemes [6, 63, 93] utilize, is ineffective against intelligent adversaries which selectively drop packets, since the probing packets are not completely indistinguishable from other data packets.
RSR	Uses forerunner packets to encourage cooperation; forerunner packets do not need to be indistinguishable from other packets. RSR successfully mitigates against selective packet dropping.

References

[1] Ns2 network simulator. http://www.isi.edu/nsnam/ns.

[2] A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Proceedings of Hawaii Int. Conference on System Sciences HICSS*, January 2000.

[3] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, pages 310–317, November 2001.

[4] G. Arboit, C. Crépeau, C. R. Davis, and M. Maheswaran. localized certificate revocation scheme for mobile ad hoc networks. to appear in Elsevier Ad Hoc Networks Journal, 2006.

[5] K. Arnold, B. O'Sullivan, R. W. Scheifler, and A. Wolrath. *The Jini specification.* Addision-Wesley, Reading, MA, 1999.

[6] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *Proceedings of the ACM workshop on Wireless security (WiSE '02)*, pages 21–30, September 2002.

[7] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, February 2002.

[8] R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16(1):87–90, 1958.

[9] J. Binkley and W. Trost. Authenticated ad hoc routing at the link layer for mobile systems. *Wireless Networks*, 7(2):139–145, 2001.

[10] M. Blaze, J. Feigenbaum, J.Ioannidis, and A. Keromytis. The keynote trust-management system version 2. Internet Request for Comments (RFC 2704), September 1999.

[11] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of the 17th IEEE Symposium on Security and Privacy*, pages 164–173, May 1996.

[12] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. Hubaux, and J. Le Boudec. Self-organization in mobile ad-hoc networks: the approach of terminodes. *IEEE Communications*, 39(6):166–174, 2001.

[13] D. Boneh and M. Franklin. Efficient generation of shared RSA keys. In *Crypto'97*, pages 425–439, 1997.

[14] D. Boneh and M. Franklin. Identity based encryption from the weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

[15] P. Bose and P. Morin. Online routing in triangulations. In *Proceedings of the 10th International Symposium on Algorithms and Computation (ISAAC'99), Volume 1741 of Springer LNCS*, pages 113 - 122, 1999.

[16] A. Boukerche, K. El-Khatib, L. Xu, and L. Korba. An efficient secure distributed anonymous routing protocol for mobile and wireless ad hoc networks. *Computer Communications*, 28(10):1193–1203, 2005.

[17] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Mobile Computing and Networking*, pages 85–97, October 1998.

[18] L. Buttyan and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications*, 8(5):579–592, 2003.

[19] C. Candolin and H. Kari. A security architecture for wireless ad hoc networks. In *Proceedings of IEEE Milcom 2002*, October 2002.

[20] S. Capkun, L. Buttyan, and J.-P. Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64, January-March 2003.

[21] D. Catalano, R. Gennaro, and S. Halevi. Computing inverses over a shared secret modulus. In *Eurocrypt'00*, pages 190–207, 2000.

[22] K. Chen, K. Nahrstedt, and N. Vaidya. The utility of explicit rate-based flow control in mobile ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference WCNC 2004*, 2004.

[23] C. Cheng, R. Riley, S. P. R. Kumar, and J. J. Garcia-Luna-Aceves. A loop-free bellman-ford routing protocol without bouncing effect. In *Proceedings of ACM SIGCOMM '89*, pages 224–237, September 1989.

[24] S. Chokhani, W. Ford, R. Sabett, and C. Merrill. Internet X.509 public key infrastructure certificate policy and certification practices framework. Internet Request for Comments (RFC 3647), November 2003.

[25] B. Chor, S. Goldwasse, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proceedings of 26th IEEE Annual Symposium on the Foundations of Computer Science (FOCS)*, pages 383–395, October 1985.

[26] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. Internet Request for Comments (RFC 2501), January 1999.

[27] C. Crépeau and C. R. Davis. A certificate revocation scheme for wireless ad hoc networks. In *Proceedings of 1st ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2003)*, pages 54–61, October 2003.

[28] C. R. Davis. *IPSec: Securing VPNs.* Osborne/McGraw-Hill, New York, 2001.

[29] C. R. Davis. A localized trust management scheme for ad hoc networks. In *3rd International Conference on Networking (ICN'04)*, pages 671–675, March 2004.

[30] S. Deering. Icmp router discovery messages. Internet Request for Comments (RFC 1256), September 1991.

[31] H. Deng, Q-A. Zeng, and D. P. Agrawal. Svm-based intrusion detection system for wireless ad hoc networks. In *Proceedings of the IEEE 58th Vehicular Technology Conference (VTC 2003-Fall*, pages 2147–2151, October 2003.

[32] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[33] W. Diffie and M. Hellman. Privacy and authentication: An introduction to cryptography. *Proceedings of IEEE*, 67(3):397–427, March 1979.

[34] J. R. Douceur. The sybil attack. In *Proceedings of the First International Workshop on Peer-to-Peer Systems*, pages 251–260, March 2002.

[35] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi. Signal stability based adaptive routing (SSA) for ad-hoc mobile networks. *IEEE Personal Communications*, 4(1):36–45, 1997.

[36] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory. Internet Request for Comments (RFC 2693), September 1999.

[37] P. Eronen, C. Gehrmann, and P. Nikander. Securing ad hoc jini services. In *Proceedings of 10th NordSec2000*, October 2000.

[38] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of 28th IEEE Symposium on Foundations of Foundations of Computer Science*, pages 427–437, October 1987.

[39] N. Ferguson. Michael: An improved MIC for 802.11 WEP. Available at: http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/2-020.zip, January 2002.

[40] G. G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. ISI Research Report ISU/RR-87-180, March 1987.

[41] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Robust threshold dss signatures. In *Proceedings of Eurocrypt '96 LNCS*, volume 1070, pages 354–371. Springer-Verlag, May 1996.

[42] M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *Proceedings of ACM 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, June 2003.

[43] P. Gutmann. Cryptlib encryption toolkit. http://www.cs.auckland.ac.nz/˜pgut001/cryptlib.

[44] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. In *Proceedings of Crypto '95 LNCS*, volume 963, pages 339–352. Springer-Verlag, August 1995.

[45] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile. Internet Request for Comments (RFC 3280), April 2002.

[46] Y. Hu, A. Perrig, and D. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of the 8th ACM International Conference on Mobile Computing and Networking (Mobicom 2002)*, pages 12–23, September 2002.

[47] Y. Hu, A. Perrig, and D. Johnson. Sead: Secure efficient distance vector routing for mobile wireless ad hoc networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02)*, pages 3–13, June 2002.

[48] Y. Hu, A. Perrig, and D. Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1976–1986, April 2003.

[49] Y. Hu, A. Perrig, and D. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceedings of the 2nd ACM Wireless Security (WiSe'03)*, pages 30–40, September 2003.

[50] Y. Huang and W. Lee. A cooperative intrusion detection system for ad hoc networks. In *Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, pages 135–147, August 2003.

[51] J.-P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001)*, pages 146–155, October 2001.

[52] J.-P. Hubaux, T. Gross, J. Y. Le Boudec, and M. Vetterli. Towards self-organized mobile ad hoc networks: the Terminodes project. *IEEE Communications Magazine*, 31(1):118–124, 2001.

[53] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt. Fire: An integrated trust and reputation model for open multi-agent systems. In *Proceedings of the*

16th European Conference on Artificial Intelligence (ECAI), pages 18–20, 2004.

[54] IEEE-SA Standards Board. IEEE Std 802.11b-1999, 1999.

[55] IEEE-SA Standards Board. IEEE Std 802.1X-2001, 2001.

[56] IEEE-SA Standards Board. IEEE Std 802.15.4-2003, 2003.

[57] IEEE-SA Standards Board. IEEE Std 802.11i-2004, 2004.

[58] Internation Telecommunication Union (ITU). Information technology - open systems interconnection - the directory: Overview of concepts, models and services, November 1993.

[59] Internation Telecommunication Union (ITU). Information technology - open systems interconnection - the directory: authentication framework, August 1997.

[60] P. Johansson, T. Larsson, N. Hedman, B. M., and M. Degermark. Scenario-based performance analysis of routing protocols fo mobile ad-hoc networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '99)*, pages 195–206, August 1999.

[61] D. Johnson and D. Maltz. Dynamic source routing in ad-hoc wireless networks routing protocols. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[62] L. R. Ford jr. and D. R Fulkerson. Flows in networks. Princeton University Press, 1962.

[63] M. Just, E. Kranakis, and T. Wan. Resisting malicious packet dropping in wireless ad hoc networks. In *Proceeding of ADHOCNOW'03*, October 2003.

[64] S. D Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the Twelfth International World Wide Web Conference*, 2003.

[65] F. Kargl, A. Klenk, S. Schlott, and M. Weber. Advanced detection of selfish or malicious nodes in ad hoc networks. In *Proceedings of the 1st European*

Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), pages 152–165, August 2004.

[66] M. Kazantzidis and M. Gerla. Permissible throughput network feedback for adaptive multimedia in aodv manets. In *Proceedings of IEEE International Conference on Communications (ICC 2001)*, June 2001.

[67] S. L. Keoh and E. Lupu. Peer trust in mobile ad-hoc communities. In *Proceedings of the 11th HP-OVUA Annual Planetary Workshop*, June 2004.

[68] S. L. Keoh, E. Lupu, and M. Sloman. PEACE: A policy-based establishment of ad-hoc communities. In *Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC)*, pages 386–395, December 2004.

[69] A. Khalili, J. Katz, and W. A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *Proceedings of 2003 Symposium on Applications and the Internet Workshops*, pages 342–346, January 2003.

[70] L. Kohnfelder. Towards a practical public-key cryptosystem. MIT Bachelor of Engineering Thesis in Electrical Engineering, May 1978.

[71] J. Kong, X. Hong, Y. Yi, J-S Park, J. Liu, and M. Gerla. A secure ad-hoc routing approach using localized self-healing communities. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'05)*, pages 254–265, May 2005.

[72] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu. Adaptive security for multi-layer ad-hoc networks. In *Special Issue of Wireless Communications and Mobile Computing*. Wiley Interscience Press, August 2002.

[73] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad hoc networks. In *Proceedings of the 9th International Conference on Network Protocols (ICNP)*, pages 251–260, November 2001.

[74] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *Proceedings of the 11th Canadian Conference on Computational Geometry*, pages 51–54, August 1999.

[75] H. Krawczyk, M. Bellare, and R. Canetti. Hmac: Keyed-hashing for message authentication. Internet Request for Comments (RFC 2104), February 1997.

[76] L. Lamport. Password authentication with insecure communication. *Communications of ACM*, 24(11):770–772, 1981.

[77] B. Lehane, L. Doyle, and D. O'Mahony. Shared rsa key generation in a mobile ad hoc network. In *Proceedings of IEEE Military Communications Conference (MILCOM 2003)*, pages 814–819, October 2003.

[78] H. Li and M. Singhal. A secure routing protocol for wireless ad hoc networks. In *Proceeding of the 39th Hawaii International International Conference on Systems Science (HICSS-39 2006)*, pages 225–234, January 2006.

[79] C. Y. Liau, X. Zhou, S. Bressan, and K-L. Tan. Efficient distributed reputation scheme for peer-to-peer systems. In *Proceedings of the 2nd International Human.Society@Internet*, pages 54–63, June 2003.

[80] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang. Self-securing ad hoc wireless networks. In *Proceedings of the Seventh International Symposium on Computers and Communications (ISCC'02)*, pages 567–574, 2002.

[81] D. A. Maltz, J. Broch, J. Jetcheva, and D. B. Johnson. The effect of on-demand behavior in routing protocols for multihop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1439–1453, 1999.

[82] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Mobile Computing and Networking*, pages 255–265, August 2000.

[83] T. S. Messerges, J. Cukier, T. A. M. Kevenaar, L. Puhl, R. Struik, and E. Callaway. A security design for a general purpose, self-organizing, multihop ad hoc wireless network. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 1–11, October 2003.

[84] M. C. Morogan and S. Muftic. Certificate management in ad hoc networks. In *Symposium on Applications and the Internet Workshops (SAINT 2003)*, pages 337–341, January 2003.

[85] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 internet public key infrastructure online certificate status protocol - OCSP. Internet Request for Comments (RFC 2560), June 1999.

[86] National Institute of Standards and Technology. FIPS PUB 113: Computer Data Authentication, May 1985.

[87] National Institute of Standards and Technology. FIPS PUB 197: Advanced Encryption Standard (AES), November 2001.

[88] R. K. Nekkanti and C-W. Lee. Trust based adaptive on demand ad hoc routing protocol. In *Proceedings of the 42nd annual Southeast regional conference*, pages 88–93, April 2004.

[89] National Institute of Standards and Technology. Secure hash standard. Federal Information Processing Standards Publications (FIPS PUBS) 180-1, April 1995.

[90] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *7th International World Wide Web Conference (WWW Consortium)*, pages 161–172, 1998.

[91] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. In *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

[92] V. D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of the 2nd IEEE INFOCOM*, pages 1405–1413, April 1997.

[93] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T Karygiannis. Secure routing and intrusion detection in ad hoc networks. In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, pages 191–199, March 2005.

[94] C. Perkins. IP mobility support for IPv4. Internet Request for Comments (RFC 3344), August 2002.

[95] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications*, pages 234–244, October 1994.

[96] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1999)*, pages 80–100, February 1999.

[97] A. Perrig, R. Canetti, D. Tygar, and D. Song. The tesla broadcast authentication protocol. *Cryptobytes (RSA Laboratories, Summer/Fall 2002)*, 5(2):2–13, 2002.

[98] A. A. Pirzada and C. McDonald. Establishing trust in pure ad-hoc networks. In *Proceedings of the 27th conference on Australasian computer science (CRPIT '04)*, pages 47–54, January 2004.

[99] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. *Commun. ACM*, 43(12):45–48, 2000.

[100] L. Reyzin and N. Reyzin. Better than biba: Short onetime signatures with fast signing and verifying. In *7th Australian Conference on Information Security and Privacy, LNCS vol., no. 2384*, pages 144–153, 2002.

[101] R. L. Rivest. The md5 message-digest algorithm. Internet Request for Comments (RFC 1321), April 1992.

[102] R. L. Rivest. SDSI – A simple distributed security infrastructure. In *CRYPTO'96 Rumpsession*, October 1996.

[103] R. L. Rivest, A. Shamir, and L. M. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[104] S. Buchegger and J. Le Boudec. Performance analysis of the CONFIDANT protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'02)*, pages 226–236, June 2002.

[105] K. Sanzgiri, B. Dahill, B. Levine, and E. Belding-Royer. A secure routing protocol for ad hoc networks. In *Proceedings of 10th IEEE International Conference on Network Protocols (ICNP)*, November 2002.

[106] J. Schneider, G. Kortuem, J. Jager, S. Fickas, and Z. Segall. Disseminating trust information in wearable communities. In *Proceedings of 2nd International Symposium on Handheld and Ubitquitous Computing*, September 2000.

[107] S. Shah, K. Chen, and K. Nahrstedt. Dynamic bandwidth management for single-hop ad hoc wireless networks. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom 2003)*, March 2003.

[108] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, November 1979.

[109] V. Shoup. Practical threshold signatures. In *Proceedings of Eurocrypt 2000 LNCS*, volume 1807, pages 207–220. Springer-Verlag, May 2000.

[110] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, pages 172–194, April 2000.

[111] I. Stojmenovic and X. Lin. Loop-free hybrid single-path/ooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032, 2001.

[112] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio networks. *IEEE Transactions on Communication*, 32(3):246–257, 1984.

[113] C.-K. Toh. Associativity-based routing for ad-hoc mobile networks. *Wireless Personal Communications*, 4(2):103–139, 1997.

[114] L. Venkatraman and D. P. Agrawal. A novel authentication scheme for ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, volume 3, pages 1268–1273, 2000.

[115] L. Venkatraman and D. P. Agrawal. An optimized inter-router authentication scheme for ad hoc networks. In *Proceedings of the Wireless 2001*, pages 129–146, July 2001.

[116] R. R. S. Verma, D. O'Mahony, and H. Tewari. Progressive authentication in ad hoc networks. In *Proceedings of the Fifth European Wireless Conference*, February 2004.

[117] W. Wang, Y. Lu, and B. K. Bhargava. On vulnerability and protection of ad hoc on-demand distance vector protocol. In *Proceedings of the International Conference on Telecommunication (ICT) (2003)*, pages 375–382, February 2003.

[118] D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM) draft-housley-ccm-mode-01.txt. Internet Engineering Task Force Internet–Draft, September 2002.

[119] M. J. Wiener. Performance comparison of public-key cryptosystems. *RSA Laboratories Cryptobytes*, 4(1):1–5, 1998.

[120] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer ecommerce communities. In *IEEE Conference on ECommerce (CEC'03)*, pages 275–284, 2003.

[121] G. Xu and L. Iftode. Locality driven key management architecture for mobile ad-hoc networks. In *Proceedings for the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, October 2004.

[122] Z. Yan, P. Zhang, and T. Virtanen. Trust evaluation based security solution in ad hoc networks. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NordSec 2003)*, October 2003.

[123] S. Yi and R. Kravits. Composite key management for ad hoc networks. In *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS 2004)*, pages 52–61, August 2004.

[124] S. Yi, P. Naldurg, and R. Kravets. Integrating quality of protection into ad hoc routing protocols. In *Proceedings of the 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002)*, pages 286–292, August 2002.

[125] B. Yu and M. P. Singh. A social mechanism of reputation management in electronic communities. In *Proceedings of the 4th International Workshop on Cooperative Information Agents*, pages 154–165, July 2000.

[126] M. Zapata and N. Asokan. Securing ad hoc routing protocols. In *Proceedings of the ACM Workshop on Wireless Security (WiSe'02)*, pages 1–10, September 2002.

[127] M. G. Zapata. Secure ad hoc on-demand distance vector (soadv) routing. INTERNET-DRAFT draft-guerrero-manet-saodv-00.txt, August 2001.

[128] M. G. Zapata. Secure ad hoc on-demand distance vector routing. *ACM Mobile Computing and Communications Review*, 6(3):106–107, 2002.

[129] Q. Zhang, T. Yu, and K. Irwin. A classification scheme for trust functions in reputation-based trust management. In *Proceedings of ISWC Workshop on Trust, Security, and Reputation on the Semantic Web*, November 2004.

[130] S. Zhong, J. Chen, and Y. Yang. Sprite: A simple, cheat-proof, credit-based system for mobile ad hoc networks. In *Proceedings of IEEE INFOCOM*, March 2003.

[131] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network Magazine*, 13(6):24–30, November/December 1999.

[132] L. Zhou, F. B. Schneider, and R. van Renesse. Coca: A secure distributed online certification authority. *ACM Transactions on Computer Systems*, 20(4):329–368, November 2002.

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