Towards Secure and Trustworthy Wireless Ad hoc Networks

Yonglin Ren

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the PhD degree in Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Due to the attractive advantages of wireless communication technologies, wireless networking and mobile computing has developed expeditiously and gained ample prevalence. Thereby, many practical applications are being designed for the use of wireless ad hoc networks in both military and civilian scenarios. However, some security concerns have arisen from such networks, especially in that misbehaving nodes pose a major threat during the construction of a trusted network. Therefore, security is one of the key challenges in wireless ad hoc networks, requiring significant attention due to their own features and concerns.

This thesis presents several computational models and security strategies for the design of secure, trustworthy networks, which are able to make rational decisions when encountering potential threats. In this thesis, we first propose a distributed network management model for secure group communication. Our approach simplifies the complexity of traditional group management and supports the inclusion of other security mechanisms for the purpose of secure communications. As a decentralized management method, trust can perform well in a dynamic and agile environment. Our proposed trust system defines the concept of trust, establishes the trust relationship between distributed nodes, involves the novel and effective computational model, and specifies a set of trust-based rules in this system for wireless nodes. We also propose a hybrid cryptosystem through the application of both symmetric and asymmetric key algorithms to provide reliable and secure protection of data confidentiality. With the design of selective encryption, uncertainty is incorporated into data encryption and the overhead spent on the data protection is significantly reduced. Thus, the communicating parties not only obtain reliable security protection, but also improve the efficiency of data communication. Through security analysis and simulation experiments, we have shown how decentralized management is useful in wireless and ad hoc scenarios, how trust provides feasible solutions for misbehavior detection, and how our proposed strategies offer security properties.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Azzedine Boukerche. If ever there was a supervisor who epitomized the phrase "going beyond the call of duty", it would be my supervisor - Dr. Boukerche. I know that I was not the easiest student to supervise, and I appreciate all your help, encouragement, guidance and financial support over the many years I have spent at University of Ottawa. Thank you very much!

The undertaking and completion of the Ph.D. program has proved to be a journey of discovery, filled with challenges and efforts, both personal and professional. Therefore, I am full of gratitude to those, who have provided me with inspiration, moral support, and technical direction. It was my good fortune to be sustained and abetted by those closest to me.

I would like to acknowledge the University of Ottawa, the Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial support during my time as a Ph.D. student.

Thanks also to Zhenxia Zhang, Richard Werner Nelem Pazzi, and Robson Eduardo De Grande, for their constructive discussion, valuable suggestions and technical support.

A big thank you out to my family - to Mom and Dad, to Weilin, Yalin, to all of my extended family. Especially, to my wife - Mian, and to my daughter - Sophia, thanks for your encouragement and inspirations over the years.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Problems of Interest 3

1.3 Summary of Contributions 6

1.4 Organization of Thesis 7

2 Wireless Ad hoc Networks **8**

2.1 Overview of Wireless Ad hoc Networks 8

2.1.1 The Features of Wireless Networks 9

2.2 Applicable Models of Wireless Networks 11

2.2.1 Mobile Ad hoc Network Model 11

2.2.2 Wireless Sensor Network Model 12

2.2.3 Vehicle Ad hoc Network Model 13

2.3 Security Challenges of Wireless Networks 13

2.3.1 The Challenges of Group Management 15

2.3.2 The Challenges of Misbehavior Detection 16

2.3.3 The Challenges of Data Confidentiality 16

2.3.4 The Challenges of Authentication 17

2.4 Threat Analysis Methodology 18

2.4.1 Risk Assessment 18

2.4.2 Types of Methodologies 19

3 SeDi: A Community-based Group Management Scheme 20

3.1 Background . . . 21

3.2 Related Work . . . 23

3.2.1 Centralized Management . . . 23

3.2.2 Clustering Management . . . 25

3.2.3 Hierarchical Management . . . 27

3.3 A Secure and Distributed Group Management Model . . . 30

3.3.1 Assumptions and Definitions . . . 31

3.3.2 A Fundamental Component in SeDi . . . 31

3.3.3 A Secure and Distributed Network Model . . . 33

3.3.4 Data Confidentiality . . . 34

3.3.5 Membership Issue . . . 34

3.3.6 Secure Key Distribution . . . 36

3.3.7 Membership Revocation . . . 36

3.4 Security and Performance Analysis . . . 37

3.4.1 Security Analysis . . . 37

3.4.2 Performance Evaluation . . . 41

3.4.3 Discussion . . . 44

3.5 Summary . . . 46

4 TOMS: A Trust Computational and Management System 47

4.1 Initiative . . . 48

4.2 Related Work . . . 51

4.2.1 Trust over the Internet . . . 51

4.2.2 Trust in Wireless Networks . . . 53

4.2.3 Price-based Schemes . . . 57

4.2.4 Linear-based Trust Model . . . 58

4.2.5 Trust-based Evaluation . . . 58

4.3 Philosophy of Trust 66
4.3.1 Trust 66
4.3.2 Trust Relationship 67
4.3.3 Trust-based Community 69
4.4 A Computational Trust Model 70
4.4.1 Initial Consideration 70
4.4.2 Model Overview 73
4.5 NEAT: A Node Evaluation Scheme with Assistant Trust 76
4.5.1 Preliminaries 77
4.5.2 Fundamental Concepts 79
4.5.3 The Mechanisms of Node Evaluation with Auxiliary Trust 84
4.6 Security and Performance Analysis 95
4.6.1 Security Semantics and Analysis 95
4.6.2 Performance Evaluation 105
4.6.3 Discussion 110
4.7 Summary 112

5 HiC: A Hybrid Cryptosystem for Data Protection 114

5.1 Motivation 115
5.2 Related Work 116
5.2.1 Cryptography over Internet 117
5.2.2 Symmetric Key Algorithm 118
5.2.3 Asymmetric Key Algorithm 120
5.2.4 Selective Encryption Algorithm 122
5.3 A Hybrid Cryptosystem with Symmetric and Asymmetric Keys 125
5.3.1 Symmetric Key and Asymmetric Key Algorithms 125
5.3.2 A Comparison of Symmetric Key and Asymmetric Key Encryption 126
5.3.3 Overview of a Hybrid Cryptosystem 127

5.4 A Probabilistic Selective Encryption Scheme . . . 129
5.4.1 Initial Consideration . . . 129
5.4.2 The Issues of Selective Encryption Algorithms . . . 130
5.4.3 The Overview of our Proposed Selective Encryption Algorithms . 132
5.5 Security and Performance Analysis . . . 137
5.5.1 Security Analysis . . . 137
5.5.2 Performance Evaluation . . . 141
5.6 Summary . . . 143

6 ToA: A Token-based Authentication Scheme 144
6.1 Introduction . . . 145
6.2 Related Work . . . 147
6.2.1 One-factor Authentication . . . 147
6.2.2 Centralized Authentication . . . 149
6.2.3 Authentication in Wireless Networks . . . 149
6.3 Two-factor Authentication . . . 151
6.3.1 Authentication Factors . . . 151
6.3.2 The Concept of Two-factor Authentication . . . 152
6.3.3 Motivation . . . 153
6.4 A Token-based Two-factor Authentication Scheme . . . 157
6.4.1 Token Generation . . . 157
6.4.2 A Two-factor Authentication Scheme . . . 158
6.4.3 Token and Key Distribution . . . 160
6.4.4 Secure Data Transfer . . . 161
6.5 Security and Performance Analysis . . . 161
6.5.1 Security Analysis . . . 161
6.5.2 Performance Analysis . . . 164
6.6 Summary . . . 167

7 Conclusion and Future Works **168**

7.1 Final Remarks 168

7.2 The Contributions of this Thesis 169

7.3 Future Research Directions 171

A List of Publications Related to Thesis **172**

List of Tables

3.1 Basic Notations and Statements 38

4.1 The Comparison of Trust Computation and Evaluation Schemes 59

4.2 Training Examples for Weather Conditions 83

5.1 A Comparison of Symmetric and Asymmetric Algorithms 127

List of Figures

1.1 Applicable Scenarios of Wireless Ad hoc Networks 2

2.1 The Necessity of Node Cooperation 9
2.2 A Military Application Scenario of Wireless Ad hoc Networks 11
2.3 A Sensor Application Scenario of Wireless Ad hoc Networks 12
2.4 A Vehicular Application Scenario of Wireless Ad hoc Networks 13
2.5 Authentication, Authorization and Access Control 15
2.6 An Overview of our Proposed Security Modules 18

3.1 Centralized Management with a Central Authority (CA) 21
3.2 The *Join* Process with *right* Certificate 25
3.3 An Illustrative Diagram of Hierarchical Management 30
3.4 An Illustrative Example of a Community 32
3.5 Average Management Overhead *vs.* Session 43
3.6 Hello Message Overhead *vs.* Session 44

4.1 The Trust-based Wirless Communication 50
4.2 Event Reporting in a VANET 64
4.3 The Schematic Diagram of Trust Evaluation 69
4.4 Graphs of Exponential Functions 71
4.5 Graphs of Logarithmic Functions 72
4.6 Graphs of Linear Functions 72

4.7 Illustrations of Direct and Indirect Trust 78
4.8 A Schematic Diagram of Naïve Bayes Classifier 80
4.9 The Application of Naïve Bayes in Trust Evaluation 88
4.10 Un-weighted Trust Evaluation and Weighted Trust Evaluation 90
4.11 Percentage of Malicious Nodes and Connectivity 106
4.12 Security Overhead based on Network Size 107
4.13 Query Overhead *vs.* Number of Assistants 108
4.14 Query Accuracy *vs.* Number of Assistants 109

5.1 Data Protection between *Alice* and *Bob* 116
5.2 Traditional Symmetric Key Update 119
5.3 An Illustration of Public Key Encryption 121
5.4 The Schematic Diagram of Selective Encryption 131
5.5 The Schematic Diagram of Symmetric Key Distribution 133
5.6 The Flow Chart of Probablity Selective Encryption Algorithm 135
5.7 Encryption Time *vs.* Network Size 142
5.8 Encryption Proportion *vs.* Network Size 143

6.1 The Relationship of *Triple-A* Methods 145
6.2 The Role Assignment during Different Authentication Processes 151
6.3 A Schematic Diagram of Two-factor Authentication 153
6.4 A Simple Handshake Procedure with TLS 155
6.5 The Scheme of Token-based Authentication 158
6.6 The Procedure of Dual Authentication 160
6.7 End-to-End Delay 165
6.8 Average Authentication Percentage 166
6.9 The Factors that Influence the Overhead of Authentication 167

Glossary of Terms

AP Access Point

ACL Access Control List

Alice An Archetypal Character (Party A)

AODV Ad hoc On-Demand Vector routing

BS Base Station

Bob An Archetypal Character (Party B)

CA Certificate Authority

DES Data Encryption Standard

D-H Diffie-Hellman

Eve An Eavesdropper

GPS Global Positioning System

IDS Intrusion Detection System

NB Naïve Bayes

MAC Message Authentication Code

MANET Mobile Ad hoc NETwork

MD5 Message-Digest Algorithm

MHS Mobile Healthcare System

MS Mobile Station

PDA Personal Digital Assistant

PIN Personal Identification Number

PKI Public Key Infrastructure

QoS Quality of Service

RNG Random Number Generator

RSU Road Side Unit

SSL Secure Sockets Layer

TLS Transport Layer Security

TTP Trusted Third Party

VANET Vehicular Ad hoc NETwork

VPN Virtual Private Network

WEP Wired Equivalent Privacy

WLAN Wireless Local Area Network

WSN Wireless Sensor Network

Chapter 1

Introduction

1.1 Motivation

In recent decades, significant technological advances in wireless networking and mobile computing have greatly motivated the expeditious prevalence of wireless devices and the rapid growth in the amount of mobile users. This is due to the important advantages of wireless communication technologies: for instance, wireless devices are portable and easily deployed, flexible wireless communication ranges satisfy different application requirements, and in some special cases, wireless devices are able to operate without pre-deployed infrastructures. Thus, a wireless environment not only reduces the cost of installation and maintenance, but also simplifies the procedure of planning and configuration. As a typical representation of wireless networking and mobile computing, wireless ad hoc networks have become increasingly popular, because traditional interconnections use wires between nodes that are fully replaced by wireless transmission methods, and also because such networks do not rely on a pre-existing infrastructure.

A wireless network differs from traditional computer networks in that it uses all kinds of portable and wireless devices to saturate its users in the wireless environments. A variety of devices including laptops, personal digital assistants (PDAs), cellular phones, blackberrys, and wireless sensors are employed for such an environment [14]. As shown

in Figure 1.1, wireless communication is prevalently used in diverse applications and security is extremely important in these scenarios. In a wireless ad hoc network, a collection of wireless nodes forms a network that does not need any pre-deployed infrastructure. Furthermore, each node has individual autonomy and all nodes can organize themselves in an arbitrary fashion. Such a network is suitable for specific scenarios in which the installation of an infrastructure is difficult or impossible due to hostile environments, high costs, or a transient period of use. Many important applications include military battlefield communications, disaster rescue operations, ad hoc meetings, mobile healthcare systems (MHS), and wireless personnel home networks, just to name a few.

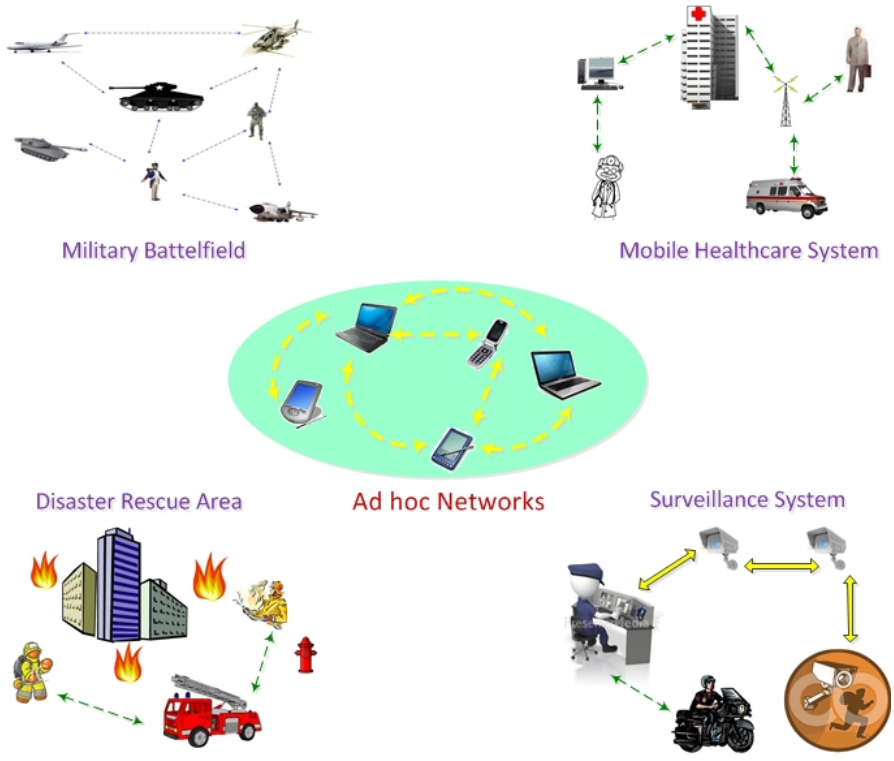


Figure 1.1: Applicable Scenarios of Wireless Ad hoc Networks

However, security is always a fundamental issue in dynamic and agile environments, including wireless ad hoc networks, as these networks are different from traditional wired networks; and thus, they have special characteristics, such as short transmission range,

shared resources, and flexible node availability, etc. These unique qualities of wireless nodes result in regular secure strategies, used in wired networks, not working well for wireless environments; this is because the nodes within wireless networks usually have low processing capabilities. In addition, a wireless ad hoc network allows arbitrary nodes to join or leave freely, and thus the topology of the network changes quickly and makes it difficult to guarantee the security of the network. Based on the peculiar needs of such networks, high-level but effective security mechanisms must be developed to satisfy these requirements.

In our work, we will concentrate on several key issues in terms of security, ranging from trust management, to data confidentiality, to authentication. Hence, we can secure the data traveling within the network. We have divided the work into several problems which we will describe in the next section.

1.2 Problems of Interest

In a wireless ad hoc network, the availability of nodes enables the network a high level of autonomy. Nevertheless, since the nodes are improvisational, data transmission has to fully rely on the cooperation among nodes. For example, when two nodes are in their mutual transmission ranges, they can directly communicate with each other; otherwise, other nodes have to cooperate to relay their exchanged information. Here, these intermediate nodes, in the network, function as routers for all other nodes, and are thus capable of accessing the relayed data. Apparently, the cooperation among nodes should be encouraged and the nodes should be managed securely, with the purpose of establishing an active and secure network. At the same time, since there are not sufficient pre-deployed infrastructures, all network activities have to rely on the cooperation of the nodes in the network, and each node is expected to be dedicated to packet forwarding for other nodes.

Wireless ad hoc networks are deployed to many important scenarios. Since these important applications have high security requirements, a wireless ad hoc network is often

faced with a diversity of security issues, such as trustworthiness, data confidentiality, authentication, and so on. Initially, the concepts of authentication, authorization, and access control were borrowed from traditional network security models, and they were combined to offer secure resource management and identity verification. In general, these solutions can work well in wired networks. However, they are not sufficiently effective for pervasive and wireless networks, because wireless networks have dynamic topology and open resources [2]. Thus, it is important to develop an operative network management mechanism, in order to effectively manage the nodes and to apply those security strategies in wireless networks. As such, data confidentiality is also regarded as one of the most fundamental concerns for the contemporary information and network security realm. Within an open and shared environment, since all data is transmitted over open wireless channels, data protection is obviously indispensable. Traditional networks employ cryptographic techniques to guarantee the confidentiality of communicated data. As an example, public key infrastructures (PKI) is widely applied in the area of data protection; whereas, due to the limited processing capacity of wireless nodes, an efficient cryptosystem is more desirable in wireless and ad hoc environments.

One direction of our work is trust-based secure management for wireless networks. A wireless ad hoc network is fully formed by wireless nodes. Therefore, this network has a high-level of autonomy and flexibility: each node determines its own availability and all nodes organize themselves arbitrarily. Similarly to traditional networks, network management services play an important role in the procedure of securing a network. In wired networks, those management services are usually achieved by a centralized management center. Nevertheless, in an ad hoc network, since there is no pre-deployed infrastructure and each node uses batteries as its power supply, it is impractical to take advantage of a centralized management methodology. Under such circumstances, the technique of trust is employed according to its distributed, historical and deterministic natures. As a result, an alternate scheme to centralized management is required to manage nodes in a decentralized manner and to secure the management of the network.

The emerging trust technique is obtaining more and more attention, because the concept of trust offers a reliable standard to measure the dependability of an expected action or object. Based on the traits of the trust technique, trust has many applications in the area of information technology. These range from Internet to electronic commerce and wireless communications. Trust has especially more important applications in wireless networking and mobile computing. This is because trust is able to evaluate an entity's credibility, to track the entity's historical behavior, and to provide incentives to encourage cooperation among various entities. According to the features of trust, it can effectively help detect misbehavior and prevent the involvement of malicious nodes in wireless communications.

It is worthwhile mentioning the behavior of each node in a wireless ad hoc network, because a node's behavior is crucial to a network; and malicious behavior can weaken the function of the network. Consequently, how to detect a node's misbehavior is significant to the security of the entire network. At present, the new technique of trust that is gaining in popularity is able to measure the role of wireless nodes and perceives risks related to the likelihood of encountering malicious behavior. As Theodorakopoulos and Baras defined in [103], "*trust* is interpreted as a relation among entities that participate in various protocols". Trust-based systems are introduced to avoid using the central trusted third authority which is not well suited to dynamic and agile environments. A trust management system can help well-behaved nodes avoid working for misbehaving nodes, in addition to assisting these well-behaved nodes to detect malicious ones. Based on the dynamic and flexible nature of the topology, trust management can easily establish a set of effective rules that reliably analyze certain suspicious nodes based on distributed and incomplete information. Obviously, trust evaluations have predetermined characteristics and can decentralize the throughput of trust computation throughout wireless ad hoc networks. In our work, we concentrate on how to calculate a node's trust, how to predict the reliability of indirect trust provided by third parties, and how to manage the trust of each node. Thereby, we are greatly motivated to design a secure and distributed

system with reliable trust evaluation, which can protect the function of data transmission securely and prevent misbehavior effectively.

Another direction of our research is the protection of data confidentiality in a wireless network. As we described before, cryptography is an indispensable component in the building up of a secure system. Without encryption and decryption, anyone can access the communicating data and even easily jeopardize it. In the modern cryptosystems, there are symmetric and asymmetric encryption algorithms. An essential concern is how to apply these encryption algorithms effectively in a wireless ad hoc network. This is because these cryptographic algorithms have different security extents and computational costs. According to the features of asymmetric key algorithms and wireless devices, it is not efficient to use asymmetric key encryption at all times in an ad hoc network, though it provides sufficient security. Thus, a secure but efficient cryptosystem should be developed, in order to reduce the overhead spent on data protection.

1.3 Summary of Contributions

The main objective for our work is to establish a set of trustworthy and distributed security mechanisms to alleviate and/or eliminate the aforementioned security concerns. Our contributions encompass several interrelated issues when promoting a solvable paradigm that aims to meet the security requirements of wireless ad hoc networks. In more detail, two models will be presented respectively that will address how to manage a network in a distributed way and how to evaluate a node's trust. In addition to clearly defining these two models, a variety of security techniques are employed to secure the data communications and to design a secure and trustworthy network.

The central problems addressed in this thesis can be summarized as follows:

- In order to deal with the issue of group management in a wireless network, we present a distributed network management method, which introduces a community-based model and thereby simplifies the complicated group management issue.

- One of our principal contributions involves the development of a solution for trust computation and management in a wireless environment. This contribution spans a few of related offerings, as follows:

  1. The concept of trust is refined and the trust-based relationship is clarified;
  2. A trust computational model differentiated from the traditional linear trust model is presented;
  3. A trust evaluation system with auxiliary trust information is proposed.

- The delicate balance between security and cost is highlighted in our analysis of the existing encryption algorithms. We reconcile both requirements and provide a hybrid cryptosystem, employing both symmetric and asymmetric mechanisms. Through the introduction of a selective encryption mechanism, we apply the traditional data protection methods to wireless communications in a balanced way.

- Based on the requriements of wireless networks, we develop a secure and reliable authentication method, which respectively makes use of the concepts of two-factor authentication and token-based authentication.

1.4 Organization of Thesis

This thesis dissertation starts with a discussion of background knowledge in wireless and mobile security in Chapter 2. In Chapter 3, we introduce the design of a community-based network management scheme. Chapter 4 describes the trust computation and management model, as well as the trust evaluation scheme with indirect trust information. Chapter 5 refers to a hybrid cryptosystem and discusses the use of selective encryption approaches in wireless networks. In Chapter 6, we outline a two-factor authentication scheme and the concept of token is employed for the purpose of authentication. Finally, Chapter 7 concludes the thesis and proposes directions for future work.

Chapter 2

Wireless Ad hoc Networks

2.1 Overview of Wireless Ad hoc Networks

One of the aims of wireless ad hoc networks is to turn the dream of allowing users to connect anytime and anywhere into reality. Due to the attractive characteristics of wireless ad hoc networks, many practical applications are being designed for use with them, including in both military and civilian scenarios [14].

A wireless ad hoc network consists of a number of autonomous wireless nodes that form networks which do not need any pre-deployed infrastructure. Since the nodes are not always stationary, the network topology may change unpredictably and even rapidly over time, and routing packets are transmitted only by relying on intermediate peers. In this type of environment, two nodes can communicate directly with each other when they are within their direct transmission ranges; otherwise, other nodes have to cooperate to relay the routing information. Thereby, these intermediate nodes work as routers for all other nodes in the network. Figure 2.1 shows that a pair of nodes which cannot directly reach each other, has to rely on other nodes to forward their messages. As a result, the cooperation between nodes is indispensable and intermediate nodes play an important role in the process of securing this network.

Both the openness of ad hoc networks and the availability of nodes make the nodes

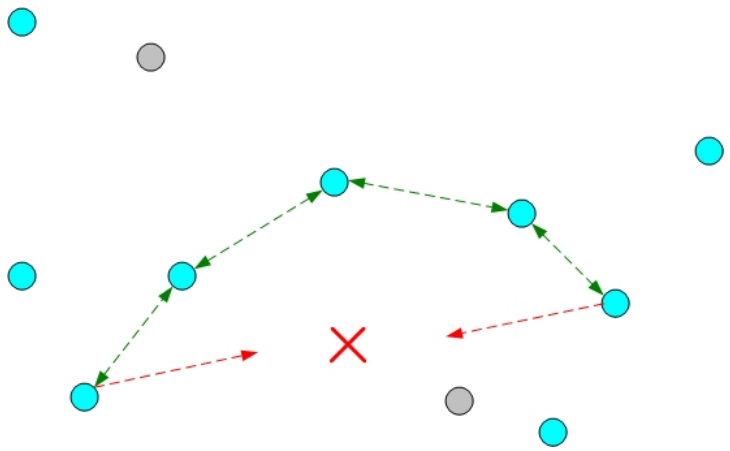


Figure 2.1: The Necessity of Node Cooperation

more susceptible to malicious behavior. Particularly, wireless devices such as PDA, laptop and smart phone, possess their own characteristics, and enable the issue of group management more complicated. In a wireless ad hoc network, each node is free to join or leave, and so the network is able to form an arbitrary topology. Thereby, the availability of nodes leads to a high level of autonomy for the network. Whereas, due to importance of the cooperation among nodes. the cooperation are encouraged and the nodes should be managed securely, in order to establish an active and secure network. In this section, we discuss the features of wireless devices and the security issues faced by wireless networks.

2.1.1 The Features of Wireless Networks

A typical wireless ad hoc network has plenty of flexibility and needs mutual cooperation for information exchanges. However, such a network also faces the following important challenges [15]:

(1) This kind of network is rapidly deployable and is able to self-organize;

(2) A wireless ad hoc network is short of pre-deployed infrastructures, so a centralized network management solution is not suitable for such an environment;

(3) Wireless devices usually use batteries as their power supply, thus they all have low processing capability and do not prefer those complicated security strategies;

(4) Due to the open and shared nature of wireless channels, the data confidentiality of exchanged messages needs to be protected;

(5) Wireless links have short transmission ranges, and thereby the cooperation between nodes is necessary;

(6) Wireless networks often experience changes in network topology, since each node has free membership in the network, leading to a further need for security protection.

Obviously, the importance of security in wireless networks has been recognized, and so a variety of security strategies are employed to protect the data communication. For instance, Bononi *et al.* [12] and Montenegro *et al.* [79] respectively introduce the scheme of intrusion detection to detect the abnormal behavior of neighboring hosts, and a solution based on Intrusion Detection Systems (IDS) is integrated with secure routing protocols to help in finding the end-to-end secure routes. Zhang *et al.* [118] develop a cluster-based detection approach. Specifically, a node will be elected as the cluster head and furthermore this cluster head will manage the IDS functions for all nodes in a network. Thus, a set of rules are adopted by the above cluster head to differentiate the normal behavior and intrusion attacks. Zhong *et al.* [122] use a clustering approach to classify traffic data in wireless networks. First, the largest cluster is calculated and other different clusters are ordered based on their distances to the largest cluster. Then they label each traffic instance as intrusion or normal by judging in which cluster this instance is and a cutoff percentage is used as a criterion between attacks and normal behavior.

2.2 Applicable Models of Wireless Networks

According to the characteristics of wireless nodes, ad hoc networks often experience changes in network topology due to the flexibility of nodes' availability or mobility. Moreover, their capacity for easy configuration and quick deployment have made ad hoc networks evolve into several variants: mobile ad hoc networks (MANET), vehicular ad hoc network (VANET), and wireless sensor networks (WSN), and so on. We respectively describe several examples of these networks below.

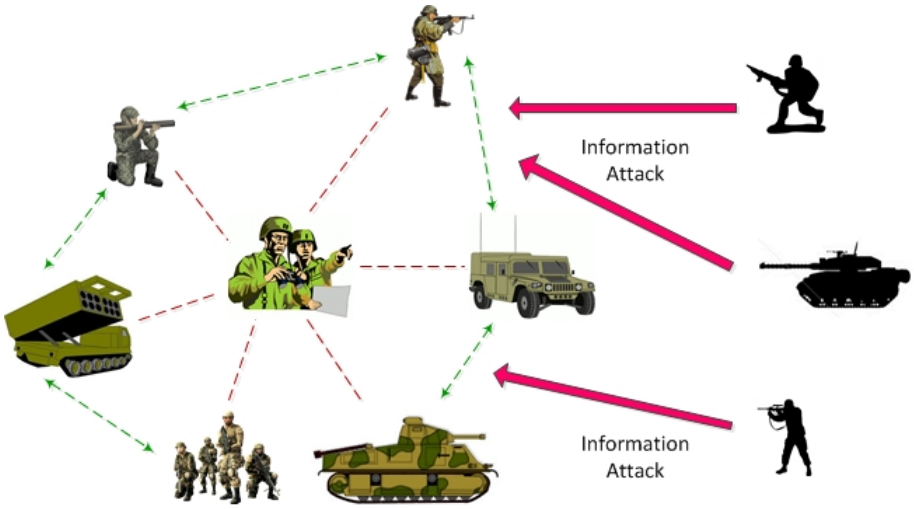


Figure 2.2: A Military Application Scenario of Wireless Ad hoc Networks

2.2.1 Mobile Ad hoc Network Model

As a typical instance of wireless ad hoc networks, a mobile ad hoc network provides flexible functionality suitable for wireless communication application within both static and dynamic topologies with increased dynamics due to node mobility or other factors. Each node in a MANET is free and independent to move in any direction, and will therefore change its links to other nodes frequently. Especially, each node is required to forward traffic unrelated to its own use. As illustarted in Figure 2.2, the mobile ad hoc networks can be applied in a battlefield scenario, in which different military professionals

can communicate using wireless devices. They form a temporary network using these wireless equipments and exchange the information about the battlefield situations.

2.2.2 Wireless Sensor Network Model

Wireless sensor networks own many characteristics from ad hoc networks. In particular, small and low-cost sensors equipped with wireless transmitters can be deployed in inaccessible areas or hostile environments, and thereby consist of a WSN. Figure 2.3 shows a WSN applied in the scenario of monitoring endangered species, in which sensors (nodes) are widely deployed in a specific area and the sensing data are sent to a few base stations. In such a WSN, their communications are often exposed in an open environment and the cooperation between sensors is indispensable. Furthermore, a WSN especially cares about the energy efficiency and maintenance of its sensors, due to the constrained resource of sensors or the unpredicted occurrence of emergency situations.

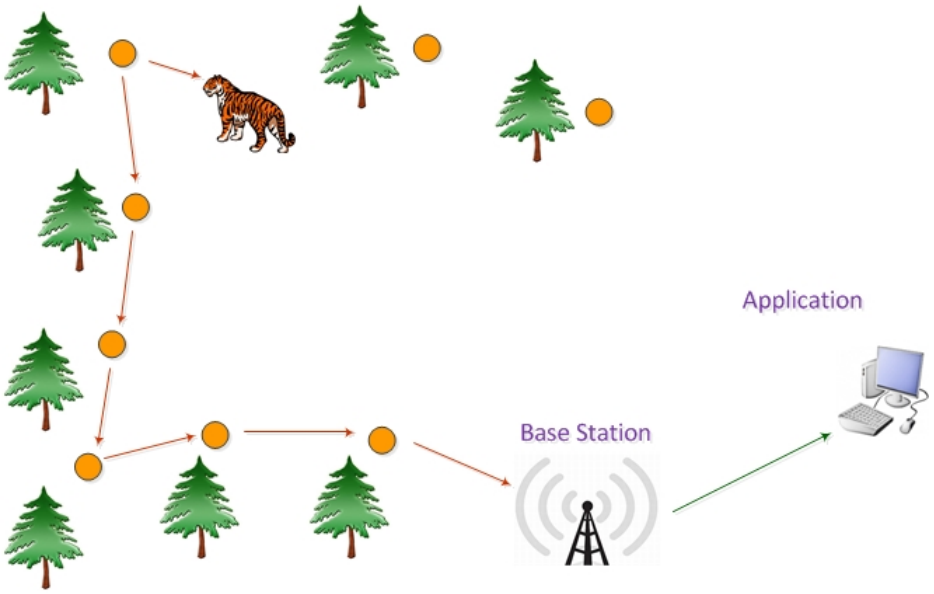


Figure 2.3: A Sensor Application Scenario of Wireless Ad hoc Networks

2.2.3 Vehicle Ad hoc Network Model

A vehicular ad hoc network is a mobile network, which takes advantage of moving cars as nodes to form an organized network. In a VANET, all participating vehicles act as wireless routers or mobile nodes, and they may move in a high speed within a wide range. Some roadside units (RSU) may be deployed at the both sides of a road. In particular, police and fire vehicles can communicate with each other for safety purposes. Figure 2.4 illustrates a schematic diagram of a VANET, in which vehicles and roadside units consist of this network. In such a VANET, the vehicles are connected by wireless links, and their communications are often exposed in an open environment, so the reliable cooperation between vehicular nodes is essential.

Figure 2.4: A Vehicular Application Scenario of Wireless Ad hoc Networks

2.3 Security Challenges of Wireless Networks

In spite of the facilitation of wireless communications and the advantages of mobile networks, it should be noticed that new risks exist in such networks, along with those

traditional threats. Though traditional security strategies attempt to be applied here, they still give rise to security concerns as follows:

- Since wireless links are the medium of data transmission, data communications rely on wireless channels instead of the use of wires. Due to the open and share natures of wireless links, it is crucial to protect their exchanged information confidentially, so that unauthorized parties cannot access the information. Through the protection of data confidentiality, many traditional threats can be prevented, including packet capture, replay attack, non-repudiation, etc.

- In traditional network security models, authentication identifies the digital identity of a user and verifies the information communicated between the sender and the receiver. In general, authentication is deemed as the primary solution to only authenticate a legal node to access correct resource and information. An important concern is how to manage the network resource in a secure way and how to prevent unwanted parties from obtaining unauthorized data and resource. Though there are many traditional authentication strategies in wired network, it is still not solved well about how to apply traditional authentication methods in wireless and mobile networks.

- As the most fundamental misbehavior in a wireless ad hoc network, packet dropping (also referred to as *black hole* attack) is vital because the wireless network needs the dedicated contributions from its participants, and most of data communications rely on node cooperation. Without the active packet forwarding, the entire network cannot perform well, and even the function of the wireless network might be paralyzed.

In fact, attacks in wireless networks are diverse: some are inherited from previous wireless technologies and the others appear as part of the new challenges of wireless networks. The above attacks are able to bring severe consequences if they are not properly

prevented. Therefore, in our research, we primarily concentrate on these threats and propose some potential solutions to deal with them.

2.3.1 The Challenges of Group Management

As we mentioned above, the unique natures of wireless devices give rise to non-trivial security concerns in terms of the management of a wireless and mobile network. The existing group management methodologies, such as, hierarchical, clustering and centralized algorithms, cannot solve the fundamental issues of group managements, because they always involve global interactions and the workload of group leaders is a bottleneck. Especially, the importance of the cluster head, group manager or CA may make them expose to potential attackers. In other words, they are much easier to become the targets of attackers due to their important status in the network. Additionally, these centralized management methods impose a lot of workload to cluster heads or group managers, so they have more processing and communication throughput than regular nodes. Therefore, from the security point of view, these traditional group management methods have quite a few security concerns and a decentralized management method is expected to deal with the aforementioned security issues.

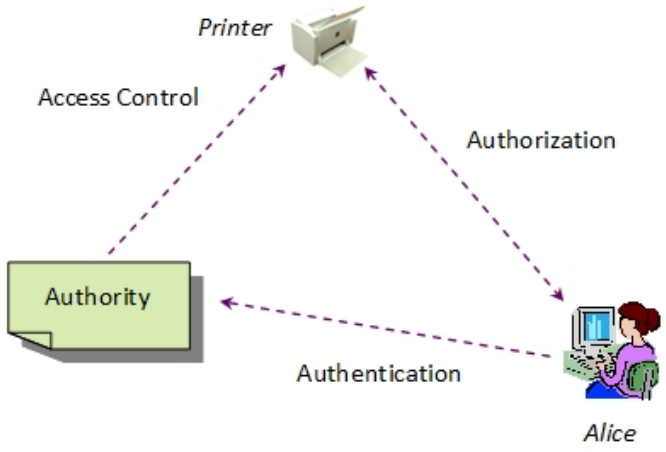


Figure 2.5: Authentication, Authorization and Access Control

2.3.2 The Challenges of Misbehavior Detection

Since the misbehavior of a wireless node is fully unpredictable, the existence of such uncertainty increases the difficulty of misbehavior detection. Because ad hoc networks lack pre-deployed infrastructures and routing packets are thus transmitted only by relying on intermediate peers, it is worth emphasizing the cooperation among wireless nodes. Nevertheless, some nodes may refuse to share their resources and even attempt to benefit from other nodes; and apparently, these nodes can jeopardize the running of a wireless ad hoc network. Based on the characteristics of wireless nodes, traditional network security solutions cannot be applied well to wireless ad hoc networks, and a wireless node with battery resource constraints cannot act also as a central management center for the entire network to provide robust security management. Therefore, some distributed strategies like trust should be introduced to detect the possible misbehavior and make decisions accordingly.

2.3.3 The Challenges of Data Confidentiality

Data confidentiality is always the principal concern of network security, and cryptography is also the primary tool to ensure the data confidentiality. At present, there are a number of cryptographic techniques in the area of information security and two representative algorithms are respectively symmetric and asymmetric key algorithms. Although these cryptographic methods are totally distinct, they are able to provide different security with different computational costs. Additionally, since wireless nodes have limited processing abilities using batteries, their computational cost spent on data protection should be tailored as well. Hence, it is noticeable to find a right tradeoff between these two cryptographic algorithms.

2.3.4 The Challenges of Authentication

In traditional network security models, authentication, authorization, and access control are combined to provide secure management. Figure 2.5 shows that *Alice* attempts to access a printer. She must first contact an authority after being authenticated. The authority checks whether *Alice* has been granted permission to use the printer. Thus, access control can be achieved by combing authentication and authorization. These solutions can work well in wired networks based on a central authority; however, they are obviously not sufficient for wireless networks, because the absence of centralized management and sometimes the scalability of these networks needs to be handled.

Currently, single-factor authentication is a classic authentication method. Whereas, for a wireless ad hoc network, as there is not sufficient pre-knowledge about a newly joining node in a network, authentication becomes extremely significant to secure the process of membership issue. Consequently, in order to eliminate the weakness of single-factor authentication and to enhance the reliability of authentication in the wireless networks, two-factor authentication may be considered as an authentication platform for wireless nodes.

In this thesis, we concentrate on the construction of a secure and trustworthy wireless network, associated with a variety of secure strategies. Figure 2.6 depicts an overview of the security modules in our work. Here, we focus on several significant security issues with the applications of emerging techniques and traditional strategies. In our design, we are aware that wireless devices do not have ample energy, and thereby over-complicated security mechanisms are not suitable. Therefore, we attempt to design simple but effective ones to protect a wireless network. For example, we consider the distributed group management methodology, the new emerging trust technique, and the effective combination of existing cryptographic algorithms.

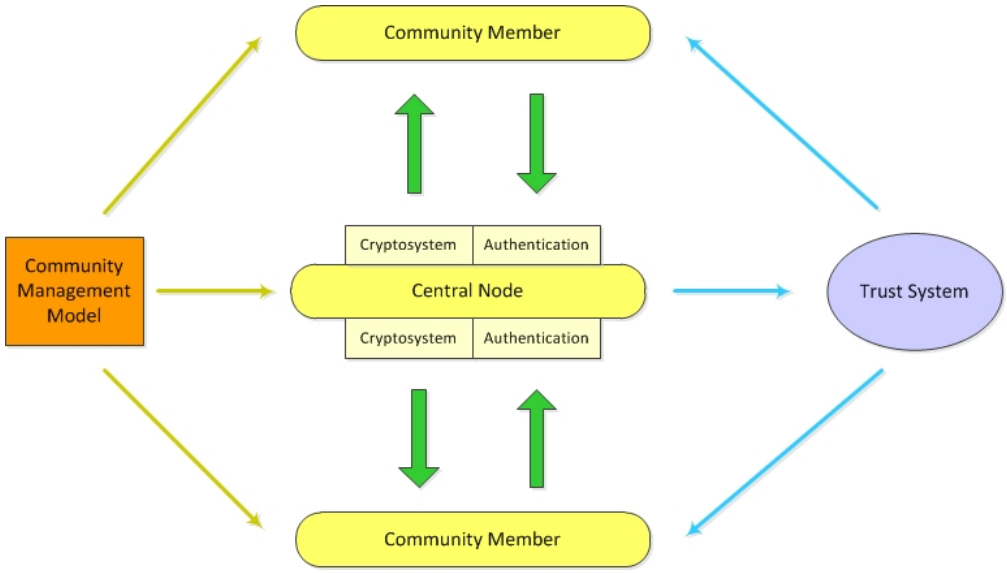


Figure 2.6: An Overview of our Proposed Security Modules

2.4 Threat Analysis Methodology

2.4.1 Risk Assessment

Risk assessment is an important step in the procedure of risk prevention; risk is probably prevented through pre-deterministic analysis and assessment. In a specific environment, risk assessment is able to determine the quantitative or qualitative value of risk, and thereby recognizes the potential threat [91]. Usually, risk assessment incorporates the concepts of *vulnerability*, *threat*, *attacker* and *risk*. *Vulnerability* occurs when one or more weaknesses exist in a given type of system, so as to allow unauthorized parties to exploit the system resources in an unauthorized way. Moreover, a *threat* occurs when *vulnerability* leaves a system open to intrusion. Thus, an *attacker*, which could be an internal or external entity, can illegally exploit this *vulnerability* and threaten other authorized users of the system for personal gain.

2.4.2 Types of Methodologies

Threat analysis is able to measure security vulnerabilities based on a given security strategy, and to provide certain guidelines for effective countermeasure devises. Major threat analysis methodologies are quantitative and qualitative in nature [41, 84]. In our research, we primarily employ the following methodologies inherent to quantitative and qualitative determination.

- *Logical Analysis*: it constitutes the security propositions and theorems, analyzes the potential security vulnerabilities, discusses a variety of security factors, and predicts the likelihood and impact of occurrence.
- *Symbolic Representation*: it takes advantage of symbolic language to demonstrate the achieved security and reliability for a given security strategy, and the prevention of possible attacks.
- *Quantitative Reasoning*: it offers an objective tool to estimate the risk posed by a threat through using the statistical probability of its occurrence.

Chapter 3

SeDi: A Community-based Group Management Scheme

Wireless ad hoc networks provide a convenient and feasible platform to realize spontaneous and improvisational computing. However, the flexibility of wireless devices also increases the difficulty of network management, because such networks do not own sufficient infrastructure with easy deployment and ad hoc nodes have autonomous availability. Under such circumstances, an appropriate group management solution is able to mitigate the effect of bad-behaved members, and to prevent the occurrence of misbehavior. In particular, secure group management ensures the functions of the entire networks and offers reliable security protection.

In this section, we first explore the security issues of group management and emphasize the importance of secure group management. Based on our proposed group principle, we furthermore present a *Se*cure and *Di*stributed group management model (SeDi). This prototype not only uses cryptographic techniques to protect exchanged information, but also employs a fully distributed manner to manage nodes in a network. Thus, each node can deal with the issue of network management individually and locally. As a result, our SeDi model avoids the use of centralized authority; on the other hand, it simplifies the traditional and complicated group concept.

3.1 Background

The concept of group management services plays an important role in the procedure of securing a dynamic and highly autonomous network. Generally, these services undertake the primary tasks of managing a network, for instance, partitioning the network's members into groups, appointing the manager for each group, distributing a group key in each group, and updating the membership of each group, and so on. In the traditional methods, group management is achieved by introducing a central management authority as shown in Figure 3.1, where all nodes are managed by an authorized center after authentication and the central authority is responsible for every aspect in terms of group management. Obviously, an ad hoc network perhaps consists of a number of nodes and the size of such a network may be scalable. Therefore, centralized management cannot deal with the flexibility of ad hoc networks well.

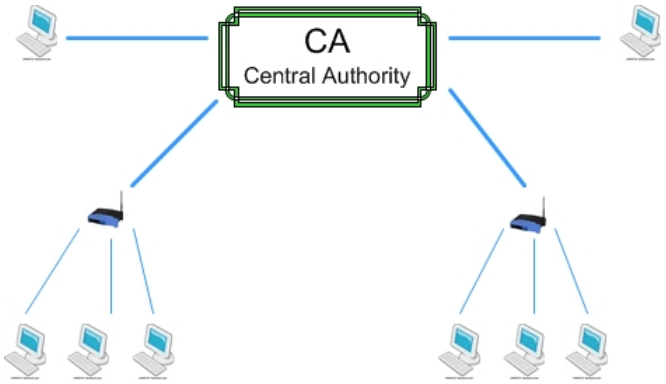


Figure 3.1: Centralized Management with a Central Authority (CA)

The importance of network management becomes prominent, due to the prevalence of personal computing devices, such as laptops, smart phones, PDAs, and so on. Unlike the traditional networks, the frequent membership changes of wireless nodes and the open transmission channel of wireless devices increase the complexity of wireless network management, and thereby secure network management in wireless networks becomes more difficult. Although attempts have been made to apply the techniques used in wired

networks, such as access control and authentication, to wireless and mobile networks as well, the effects of these strategies are unsatisfactory due to the features of wireless networks. Consequently, a high-level secure group management solution is desired in wireless ad hoc networks.

With respects to membership management, the issuer of memberships has to manage all aspects of its issued memberships, such as, membership distribution, membership update, membership revocation, and so on. Due to the flexible availability or mobility, a wireless ad hoc network usually provides free membership to those nodes which remain online or get offline freely. Apparently, free membership in such a network brings many concerns to the security of group management: a node is able to arbitrarily get offline or online from a group. At the same time, since there is not a pre-deployed infrastructure, all network activities have to rely on the cooperation of the nodes in the network, each node is expected to be dedicated to the packet forwarding for other nodes. Consequently, membership management becomes complicated in wireless environments.

In particular, the memberships need to be kept as fresh as possible; otherwise, these memberships may come to be stale or be exploited by compromised nodes. Usually, a central trusted third party (TTP) can accomplish these membership management tasks well in wired networks, but this is not practical in a wireless ad hoc network since there is not any pre-deployed infrastructure to serve as a TTP. Hence, it is extremely important to explore how to manage such a wireless and ad hoc network in a secure and effective manner.

The aforementioned central group management strategies are not suitable to dynamic and agile ad hoc environments, and new solutions are desired in wireless networks. One direction of our research is exploring what kind of network model is adopted to manage nodes, how to stimulate all nodes behave well, and therefore form a secure network. First of all, we study the issues faced by network model establishment and discuss the necessity of secure network management, through the review of some related literature.

3.2 Related Work

Recently, some research initiatives have been developed to address the problem of group management in distributed and wireless environments [54, 61, 85]. Traditional network management schemes are mainly classified in three categories: 1) centralized management, in which all nodes must obey the management from a central authority; 2) clustering management, in which all nodes are clustered into different groups and each group has its own clusterhead to control the whole group; 3) hierarchical management, in which a network is divided into different layers based on some predefined rules, and some nodes are elected as leaders to manage their own layer. In the following sections, we will respectively study the suitability of each category.

McHugh and Michael [76] investigate the issues faced by group management, including group membership, access control, assurance, key management, and so on. Their research indicates that secure management of the large scale groups cannot be simply treated as the issue of key management. Membership management should consider manifold factors, such as the trust of communicating participants, the nature of the members, each member's transmission pattern, etc. The authors also discuss the advantages of two fundamental modes of group structure, hierarchical and flat. Labonte and Srinivas [60] present a case study of secure group management service (SVPN) based on a virtual private network. Their proposed group management strategies explore whether or not an invitation is necessary for accessing the existing session, when the service provider considers whether a new user should be allowed to join the session.

3.2.1 Centralized Management

Centralized management is still the main approach in the current work of group management. For instance, Zhou *et al.* [123] employ a certificate authentication server in their group management architecture to provide authentication service for other servers and members. Autran and Li [6] explore a new management approach to fulfill the

network management service, which takes advantage of the mobile agent technique to overcome the scalability of a network and a mobile agent migrates from the server to those visited hosts. The agent can implement regular tasks and use a credential to all of its visited hosts. In addition, middleware service is deemed a solution to solve the problem of effectively managing the nodes in MANETs. Bottazzi *et al.* [13] study the issues of group management middleware. This middleware is responsible for managing the resource sharing in a group, monitoring the availability of group members, and assigning the right of resource access.

A Sevice-oriented Multicast Group Management Protocol

Kaya *et al.* [54] propose a secure multicast group management protocol by virtue of using a *right* certificate service. In the design of group management protocol, those nodes which expect to join to the multicast group have to initiate a join request through the closest neighbor already within the group. Then the initiating join request is broadcast within a limited range. The protocol primarily consists of the *join* and *leave* processes:

(1) *join* process: the new node first broadcasts its group join request. Then the receiving neighbors process further with a group join reply, and finally they mutually authenticate and establish a link key;

(2) *leave* process: when a node decides to leave the presently group, it will inform its downstream and upstream node to allow them reconstruct a connection without the leaving node.

The whole process is integrated within a session, in which a manager that is responsible for authenticating joining members, checking service *right* certificates, maintaining information about attached group members, and so on. Figure 3.1 shows the *joining* process of a node with *right* certificate, and this node A has to contact with the corresponding *right* certificate service center first to obtain a valid *right*. The *right* certificate

service will verify that a node is authorized to join a group, suggesting that an authorized node can prove to other authorized nodes that it has the evidence from the *right* certificate service to access the multicast data stream.

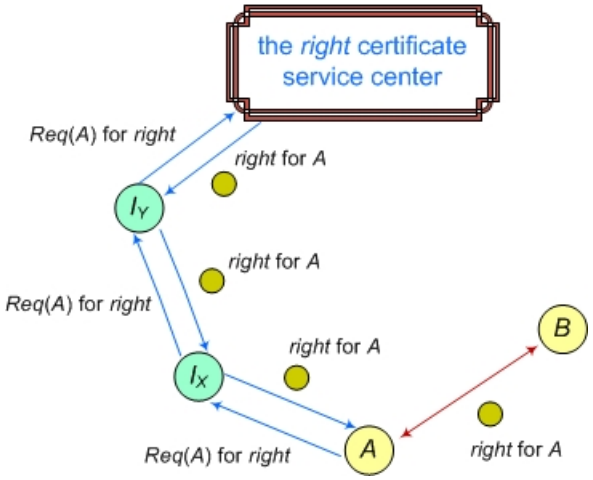


Figure 3.2: The *Join* Process with *right* Certificate

3.2.2 Clustering Management

Clustering algorithms are similar management methodologies to hierarchical-based ones, and they are widely applied in WSNs and other networks. Generally, sinks or base stations are deployed to act as the central management centers and are responsible for the collection of detected data, the monitoring of neighboring sensors, the distribution of each node's key, etc. Bandyopadhyay and Coyle [8] design a distributed clustering algorithm for a WSN and define two roles in the WSN: volunteer clusterhead and forced clusterhead. A sensor can advertise itself as a volunteer clusterhead within its direct transmission range; however, when any other sensors join the advertised clusterhead in k hops, this clusterhead will become a forced clusterhead. By means of such a hierarchical structure, the WSN can self-organize its sensors to various clusters. Chan *et al.* [24] discuss the issue of key pre-distribution in a WSN. Their solution is primarily deployed on a single cluster and the distribution of keys takes place inside the cluster. For the

clusterhead and its members, they take advantage of random methods to set up keys for each member and mutual authentication between a pair of sensors is also achieved by a randomly generated pairwise of keys. Liu and Li [70] design a mobile cluster protocol which includes mobile clustering and cluster maintenance algorithms. In the cluster establishment phase, each node sends a control message to rapidly deploy the cluster and only those nodes, which do not correctly receive this control message, are acknowledged; in the cluster maintenance phase, the cluster will update the cluster's condition only when a node moves out of the current cluster. Wang [108] investigates some important issues for a hierarchical wireless sensor network, and especially focuses on how to determine the amount of clusterhead and how to deploy them in a WSN. Some interesting factors which affect the assignment of clusterhead are discussed, such topology control, energy efficiency, etc.

A Location-dependent Group Membership Scheme

Liu *et al.* [69] present the design of a group management service for mobile computing and the attributes of group membership are investigated. Ad hoc groups are formed by neighboring nodes in a geographical region based on two rules:

(1) location-unaware groups: for those nodes which are present in an ad hoc group, they do not depend on the geographic region or location, instead of the functional properties supported by the group, for example, business-to-consumer interactions in malls, airports and trade fair communities;

(2) proximity-based groups: groups are formed by proximate nodes within a given geographical area, which can be dedicated to resource sharing with QoS properties, such as enhanced data availability or real-time processing.

Subsequently, a lot of factors are taken into account as the constraint attributes of membership, such as location, group scale, trustworthiness, QoS awareness, etc. More-

over, they present the design and implementation of a generic group service in a MANET, which primarily has three functions in group management:

(1) group member discovery: a node may broadcast a *disc* message to its neighbors to discover the nodes which can be grouped together to form a group; or other nodes can initiate a *join* message to an existing group to indicate their intentions for the group;

(2) group initialization: Each group has a node named *group leader* responsible for a) managing the group dynamics while enforcing group constraints, b) inter-group communication;

(3) group dynamic management: in this phase, it focuses on leader selection, and some criteria have been proposed for a) highest degree, where nodes with the maximum number of neighbors are selected as group leader; b) extrema-id, where the node with the smallest or greatest *id* is assigned as the group leader; c) node weight, which is an integrated metric for evaluating the suitability of a node as a group leader. It can depend on various factors such as the resource richness and the neighbor amount.

3.2.3 Hierarchical Management

Hierarchical structures greatly alleviate the problems of large-scale group membership because the organizational method avoids direct communications that cross many levels of the hierarchy. Thus, hierarchical grouping is often used in network management; this occurs especially in the case where some nodes are chosen as the heads of their own group, to manage other nodes and to maintain the running of the group. Pei *et al.* [85] propose a group mobility model, in which nodes are partitioned into groups and each group has a conceptual center to represent the group's motion. Each group is viewed as an entirely individual entity and the group's motion is multifarious, including location,

speed, direction and others. Bohge and Trappe [11] adopt a three-layer architecture which classified all nodes in a network to three different levels according to their resource limitations: *high* power, *medium* power and *lower* power. An incoming node will be classified as well based on its computational resource.

MOVI: a Distributed Network Management System

Lee [61] designs a 3-tier group management protocol for distributed systems. In their works, the hierarchical concept is well applied in the partition of a network. In the three level architecture, the first level is the basic one which only manages hosts locally, while the second level supports the inter-group interactions. In the third level, it provides global group management. Their functions are as follows:

(1) *Local Group Manager* (LGM), which manages one or more members in one node's neighborhood;

(2) *Group Communication Daemon* (GCD), which supports the overlapped grouping facility in a single domain;

(3) *Global Group Manager* (GGM), which manages LGMs and GCDs, and provides group communication over the whole network.

In this way, an LGM is able to manage multiple members which reside in a local host, a GCD provides the overlapped group management function, and the GGM manages all LGMs in a network. The actual group communication is performed by three steps: the first step is between members and LGM, the second between LGMs and a GCD, and the last between GCD and GGM.

A Role-based Hierarchical Connected Dominating Set (CDS) Algorithm

A hierarchical connected dominating set (CDS) architecture is designed for scalable operation of a large scale wireless ad hoc sensor network. A role-based self organization

algorithm is proposed in [58] to extend the hierarchical CDS architecture, in which nodes are assigned different roles depending on their connectivity and sensing capabilities, such as routing and sensing, etc. Totally, three categories nodes are designated based on their roles: 1) sensing collaborator; 2) sensing coordinator; 3) backbone nodes.

Specifically, all nodes in this network are assumed to collaboratively sense target events, so they are all sensing collaborators. Whereas, backbone nodes will fulfill the routing and backbone functions. They are supposed to support a network-wide routing functionality for both application specific sensing queries and the sensing data gathered by the sensors. For the role of sensor coordinators, they take the responsibility of not only coordinating the sensing activities in their sensing zone, but also aggregating and forwarding the information to any remote data sink or base station.

In order to attribute sensing activities, the authors take advantage of two sensing metrics known as the sensing proximity value (SPV) and the cumulative sensing degree (CSD) respectively to partition the sensor network into several sensing zones. These sensing zones individually act as an aggregation area which is comprised by sensor nodes collaborating to achieve a common sensing objective, as follows:

(1) SPV is used to measure how close a sensing cell is to a particular sensor;

(2) CSD is described as the degree of cumulative fault tolerance sensing for a common area monitored collaboratively by some sensors.

Additionally, those nodes with a common sensing objective are clustered into an individual zone and coordinators are elected to act as leaders for their respective zones. The sensor coordinators not only collaborate among members in their sensing zone, but also support network reorganization and maintenance.

Figure 3.3 is a schematic diagram of hierarchical management. Nevertheless, traditional management strategies are no longer suitable to dynamic and mobile environments; and flexible management methods are desired. Especially, the formed groups via the above schemes usually involve intermediate members in group management and the

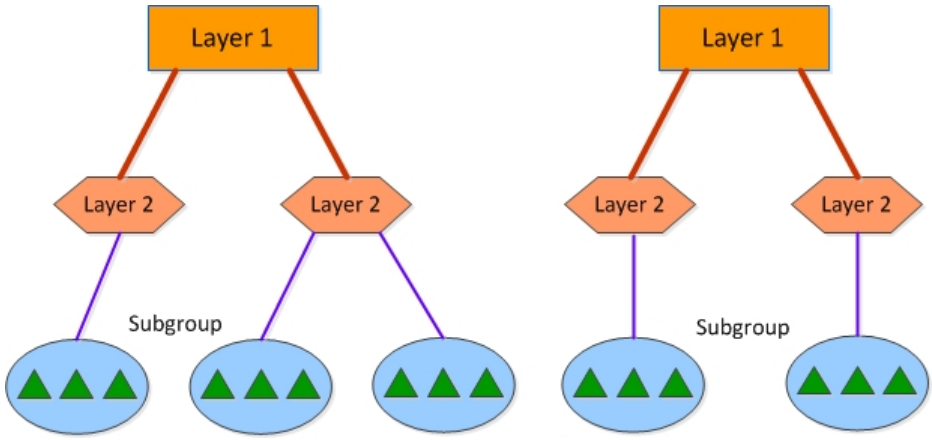


Figure 3.3: An Illustrative Diagram of Hierarchical Management

election of group leaders (or clusterheads) is limited by some physical conditions, such as computational capability, location information, etc. In addition, the workload of group leaders becomes extremely heavy when the size of network increases, and also they are easily attacked by malicious nodes according to their important status in the network. In sum, the above management mechanisms do not work well within wireless and mobile networks due to scalable computation and increasing communication costs.

3.3 A Secure and Distributed Group Management Model

In this section, we propose a novel community-based management framework that takes advantage of a variety of security strategies to manage the members of a community. First, we introduce a community prototype as the basis of group management, and describe every aspect of our network management model to build up a secure and reliable wireless ad hoc network. Our model aims to protect the confidentiality of the communicated data. Different security strategies are applied then, including trust management, symmetric key algorithm, authentication, and so on. In this way, our group management

framework is able to guarantee the security of a group, as well as simplify the issue of group management.

3.3.1 Assumptions and Definitions

We first introduce preliminary assumptions and concepts before elaborating on these details.

- The links between wireless nodes are always bi-directional.
- Every wireless node has enough computational power to finish these operations.
- There is a trusted certificate authority (CA) outside of the wireless ad hoc network that is only responsible for issuing the public keys and private keys for the wireless nodes inside the network.
- Some malicious nodes may exist in the network, which are not willing to cooperate for data transmission but they are not collude each other. That is, each malicious node behaves independently.

3.3.2 A Fundamental Component in SeDi

The concept of "group" is complicated and not easily managed since it classifies the nodes in a network into different clusters based on certain rules [58, 76]. As we have already discussed, hierarchical and clustering algorithms cannot solve the fundamental issues of group managements, as they always involve global interactions and the workload of group leaders is a bottleneck. From the security point of view, these group leaders are easier to become the target of attackers due to their importance. Consequently, we introduce the principle of an independent and local community, and the definition is as follows [16]:

"*A community is defined as a central node and all of its one-hop neighboring nodes as a community*"

Thus, we break down the complicated issue of "group" to the most basic and simplest scenario, in which each node of a network is a community and also each node is the central node of its own community. In each community, the central node only manages its direct neighbors. Additionally, both trust and cryptographic techniques are employed to protect the confidentiality of communicated data and to secure community management, which will be elucidated in Sections 4 and 5.

This one-hop community model is significant because it is the basis of our distributed group management framework. Based on the prototype of one-hop community, every node in a wireless network forms such an independent community, and also is the only central node in that community. That is, a node has multiple roles in the network: on the one hand, a node is the central node in its own community; on the other hand, this node may be the member of other nodes' communities as well. Figure 3.4 shows an example of a one-hop community, in which central node C has three neighbors: A, B and D. Node M is regarded as an untrustworthy node in community C, and so is not included in the community C.

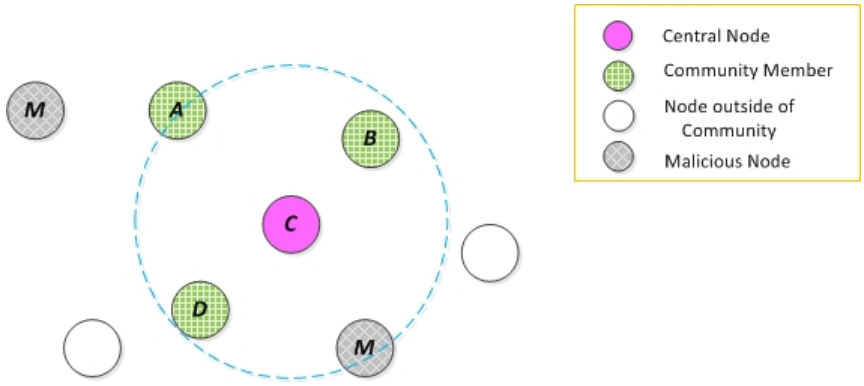


Figure 3.4: An Illustrative Example of a Community

In our SeDi network model, all communications in a one-hop community are protected by cryptographic techniques. The confidentiality of communicated data is achieved by virtue of a combination of symmetric and asymmetric keys. At the same time, as an important management method, the technique of trust is able to help identify malicious

nodes and thereby exclude them from a presently healthy network. The trust-based community model can manage nodes dynamically and the nodes' activities are evaluated effectively in a fully distributed manner. As a result, such a model alleviates the problems arisen from "group" and minimizes the average overhead spent on interactions among groups. Apparently, our trust-based community principle owns the less complexity and improves the network's security as a whole.

3.3.3 A Secure and Distributed Network Model

We have presented the definition of the basic community component in the above section. Thereby, our secure distributed group model is built up with such one-hop communities throughout the entire network. As we described in Section 3.3.2, a node that is a central node will include its one-hop neighbors into its community, in which the authentication between it and these members is achieved using cryptographic techniques, and all members are managed by means of trust evaluation. In such a community, mutual relationships between a pair of communicating parties are important to establish a trusted community. They represent if the node trusts another node to perform the intended operation; the established trust relationship between these two nodes is deemed to be reliable from the communicating initiator's point of view. In this way, the communications in the community are established over such trustworthy relationships. For instance, for the central node C and one of its members N_i, C is the sender and N_i is the forwarder. If the member N_i successfully forwards a packet for C, then N_i is considered to be an honest node for C, and C thus increases its trust value for N_i's good behavior; otherwise, if N_i lies about or exaggerates its contribution to routing, then N_i is deemed a suspicious node that will be penalized to decrease N_i's trust value accordingly. In terms of trust management, we will elaborate it in Section 4.

3.3.4 Data Confidentiality

Cryptography has been widely employed to protect the confidentiality of communicated messages, and our proposed community also adopts two major categories of cryptographic techniques, symmetric and asymmetric encryption algorithms, to ensure the security of exchanged information. Due to the open channels of wireless transmission, all communication in our proposed one-hop community needs to be protected. Nevertheless, because of the constrained computational power of wireless devices, it is not realistic to encrypt all information using the public key algorithms. Thus, all official data communication between the central node and its members will be encrypted through symmetric key algorithms, and in the meantime, asymmetric keys algorithms will be applied in some important scenarios. Regarding the issue of cryptosystem, we will present more details in Section 5.

Algorithm 3.1 The formation of a community C

$C \succ Center$ — $\#C$ sets itself as the center
for $i = 1 \rightarrow n$ **do**
  $C \rightarrow HELLO(i)$
end for — $\#C$ broadcasts HELLO message
$N_i^C \leftarrow HELLO$
if $N_i^C \vdash HELLO$ **then**
  $N_i^C \xrightarrow{Joining} C$ — $\#N_i^C$ intends to join C
else
  $N_i^C \dashv HELLO$ — $\#N_i^C$ does not intend to join C
end if

3.3.5 Membership Issue

In comparison with traditional trust management schemes [6, 61], each node has its own community centered at itself in our community-based management model. When a node wants to join an existing community or moves into the neighborhood of a central node, it will first compose a joining request message and send it to the central node. In this request message, the newly joined node informs the central node of its public key for

the authentication between them. After receiving the request message, the central node assigns an initial trust value to the newly joined node and simultaneously generates a secret key for it. A joining reply message will be composed to contain the necessary information and then be sent to the newly joined node to indicate the successful membership issue. In order to distribute the secret key securely, the central node will encrypt it using the public key of the intended neighboring node before sending it. Here, we formally describe the procedure of community formation.

In our one-hop community model, the central node generates different secret keys for different members. Thus, each member has a different secret key for communicating with the central node and all information exchanged is encrypted using the corresponding unique secret key. Due to the flexibility of nodes, whenever a node leaves or joins the neighborhood of the central node, the central node will keep its list of neighboring members as fresh as possible. Except the secret key generated for each member, the central node also has a group key available to all of its members, which is distributed together with each member's unique secret key. However, the group key is only used if the central node needs to broadcast some community messages to all of its group members. When a central node needs to communicate with an individual member for the purpose of routing or traffic relay, it still uses the unique secret key for the node-to-node communications. The following algorithm shows the procedure of a node's joining.

Algorithm 3.2 A node N joins the community C

$C \leftarrow PK_N$ #C receives the public key of N_{new}^C
for $i = 1 \rightarrow k$ **do**
  if $ID_N \in BL$ **then**
    $Decline\ N_{new}^C$ #if N_{new}^C is in BlackList, decline it
  else
    $Accept\ N_{new}^C$ #otherwise, accept it
  end if
end for
$Initialize\ Trust(N_{new}^C)$
$N_{new}^C \leftarrow SK_N^C$

3.3.6 Secure Key Distribution

In addition to the key distribution at the first time, when the secret key is generated for a newly joined node in a community, the public key of the newly joined node will be used for the distribution of its secret key at all time. For the members of a central node, different members will have different secret keys as the cryptographic tool for the communication between the members and the central node. At the same time, these keys will be updated periodically, so as to be kept as fresh as possible and avoid the exploitation of stale keys. When it comes to the update of a secret key, one member's public key is always used to encrypt the updated secret key, and then the encrypted update key is sent to the corresponding member for its key update. Throughout the whole process of data transmission, the public key and the unique secret key are combined together to provide effective data protection.

3.3.7 Membership Revocation

In a community, the central node will independently set a trust threshold and any neighboring members that cannot meet the trust requirement will be excluded from this community. Suppose that a central node observes malicious behavior and detects malicious or uncooperative nodes. Subsequently, these nodes trust will decrease accordingly based on a trust computation model. Similarly, when the central node finds that its members contribute to forward its packets, these members' trust will increase as well. Thus, the central node will track the behavior of a suspicious node and observe the changes in its trust. Here, we consider that the uncooperative behavior of a node may be caused by some objective reasons, such as synchronization, traffic congestion, interferences, etc. The nodes trust does not decrease directly to the trust threshold, but gradually lessens accordingly. However, once their trust falls below the community's trust threshold, the central node will first revoke the secret keys issued to the malicious nodes for their authentications and communications. Furthermore, it will take the malicious nodes out of

its community. As such, the independently set trust threshold by the central node will be used to restrict the entrance of unqualified nodes. The algorithm below simply describes the process of membership revocation.

Algorithm 3.3 The revocation of a node's membership

$C \succ Thres_C$ $\quad \#C$ sets a trust threshold
for $i = 1 \rightarrow m$ **do**
  if $Trust(i) \leq Thres_C$ **then**
    $SK_i \notin C$
    $ID_i \notin ML$ $\quad \#C$ removes i from member list
  end if
end for

3.4 Security and Performance Analysis

In this section, we study the characteristics of our proposed SeDi scheme and analyze its security properties through a series of proof of correctness. In addition, we also carried out a set of experiments within a wireless and mobile environment to evaluate its performance accordingly.

3.4.1 Security Analysis

Here, we provide a formal analysis of our trust-based management framework and prove that the network security can be achieved effectively within our framework. The methodology related to security analysis [40, 82] is utilized here. To simplify the security analysis, we focus only on some primary stages of our secure strategies, such as membership issue and revocation, data transmission, key distribution, etc. The underlying notations are as follows [21] in Table 3.1.

The analysis starts from the most basic behavior based on our community prototype. First, we prove that the process of a new node joining a community is secure.

Table 3.1: Basic Notations and Statements

C	The central node of a community
N_i^C	One of the neighbors in community C
PK	The public key of a certain node
SK	The secret key generated by C for neighbor N_i^C
$P \succ A$	A principal P_i possesses a specific action A
$P_i \xrightarrow{X} P_j$	The principal P_i transfers X to another principal P_j
$P_i \longleftrightarrow P_j$	An interaction between P_i and P_j with an action A

Theorem 1. *The community model provides protection to the process of a new node joining a community.*

Proof. (1) At the joining request phase, N_{new}^C is a node that attempts to join the community C, and it first gives its public key PK_N to C when entering C's neighborhood, like $N_{new}^C \xrightarrow{PK_N} C$.

(2) Then at the joining reply phase, C will send a secret key SK_N to N_{new}^C using the public key PK_N of the newly joined node, like $N_{new}^C \xleftarrow{PK_N[SK_N]} C$. As a result, C is able to communicate with N_{new}^C through the combinatorial protection of symemtric key and asysmmtric key.

Theorem 2. *The trust requirement is able to provide reliable multicast transmission within a community.*

Proof. (1) $C \succ T_r$: When the central node C wants to transmit a message in its community, it will first choose a trust value as the message's trust requirement T_r.

(2) $C \succ Select(N_i^C) \mid Trust(N_i^C) \geq T_r$: Only those neighbors whose trust values satisfy this trust requirement ($Trust(N_i^C) \geq T_r$) are chosen to receive this message. If the trust value of N_i^C is greater than or equal to T_r, the neighbor N_i^C is selected for the succeeding multicast.

(3) $C \succ Send$: Thus, only those neighbors which are filtered by the trust requirement are qualified for packet forwarding. They will receive the packets from the central node and forward them as expected, due to their satisfactory trust.

Theorem 3. *SeDi takes advantage of a fully distributed method to achieve the purpose of group management.*

Proof. Because we adopt a one-hop community model as the basis in our group management scheme, each node N_i^C in a wireless network is viewed as a group (community), which is the most basic and simplest case in the area of group management. At the same time, each node is the central node C in its own community, and the central node C can manage its one-hop neighbors N_i^C without any packet forwarding from intermediate nodes I_i. Thus, the entire network consists of the smallest unit, the one-hop community, and there is not any centralized management authority in the network to coordinate the information exchange or synchronization among different groups. Hence, the simplest group unit (the one-hop community) is pervasively distributed in the whole network with the deployment of wireless nodes, and the wireless ad hoc network is really managed in a completely distributed way.

Theorem 4. *Direct management communication prevents the possible attacks from indirect intermediate nodes, and manages neighbors via trust-based evaluation.*

Proof. As described above, the use of intermediate nodes not only can easily introduce attacks, because intermediate nodes I_i may put the process of management and communications at risk by means of jeopardizing the packets that they are forwarding, but also can increase the delay of communications, as those management data need be exchanged globally. However, the proposed framework does not introduce any indirect communication and all information exchange is monitored under the central node C, since its neighbors N_i^C are all in its direct transmission range, and thereby C can overhear the behavior of its neighbors N_i^C easily. Without intermediate nodes I_i, those attacks such as packet dropping, packet modification, malicious ID [17], etc., can be prevented effectively. Additionally, the central node C can directly manage its neighbors N_i^C based on trust-based technology, which provides a reliable criterion to evaluate a neighbor's behavior. For example, if a neighbor N_i^C successfully forwards packets for the central node C, C will increase N_i^C trust ($Trust(N_i^C) \uparrow$) accordingly; whereas, if the neighbor

N_i^C refuses the forwarding request from C, C will decrease ($Trust(N_i^C) \downarrow$) as well. Thus, without forwarding packets via intermediate nodes, the proposed framework can improve the security and effectively evaluate its neighbors.

Theorem 5. *The independent secret key avoids disclosing the same secret key to other neighbors, when communicating between the central node and its neighbors.*

Proof. In a one-hop community, the central node C will assign an independent secret key SK_i to each of its neighbors N_i^C, so there is not any other neighbor to share the same secret key in the same community. Thereby, this avoids the situation in which the encrypted data is deciphered by other neighbors which have the same secret keys. When the central node C wants to communicate with one of its neighbors N_i^C, it will encrypt the communicated message $MSG_{(C,N)}$ using the unique secret key SK_i only issued between it and the corresponding neighbor N_i^C. During the process of transmission, even if other neighbors N_j^C can intercept the encrypted message $SK_i[MSG_{(C,N)}]$ or $SK_i[MSG_{(N,C)}]$, they cannot still decrypt the message $SK_i[MSG]$ encrypted by SK_i since the corresponding secret key SK_i is confidential in this community C. As a result, the mechanism of one neighbor one secret key can effectively prevent data disclosure incurred by sharing the same secret keys with other nodes.

Theorem 6. *The proposed community management framework enhances the network security, simplifies the complexity of management, and improves the effectiveness of group management in a network.*

Proof. Obviously, the introduction of intermediate nodes I_i increases the uncertainty and complexity of group management, since the successful data communications depend on the reliability and security of these intermediate nodes. Unlike most of the existing group management schemes which adopt hierarchical structure or clustering technique, our proposed one-hop community does not utilize any structural management and allows each node to only manage its own local area. In particular, each central node C just communicates with its one-hop neighbors N_i^C without the necessity of intermediate nodes

I_i. Thus, our one-hop community prototype provides the simplest paradigm for group management.

Based on our proposed one-hop community model, our group management framework is able to avoid the use of centralized management mode and to manage a network in a completely distributed way. At the same time, all one-hop communities do not need to spend overheads on global interaction. Since the central node in each community only manages its local one-hop neighbors by means of the trust-based technology, many attacks arisen from intermediate nodes can be prevented effectively. Therefore, the proposed framework can enhance the security of the entire community and also improve the effectiveness of group management.

3.4.2 Performance Evaluation

For this section, we carried out an extensive set of simulation experiments based on the Network Simulator *ns-2* [49], in order to evaluate the performance of our system and to observe its behavior. The experimental environment was constructed within a rectangular area of $670m \times 670m$ and was comprised of N nodes ($N = 30$). These nodes moved around at a maximum speed of $5m/s$, based on the random waypoint model where each node randomly chooses its initial position, moves at a speed distributed randomly between 0 and some maximum speed, and remains stationary for a given period of pause time [18]. At the same time, we set up the pause time at 20 seconds before each node could move to its next destination, and we set the transmission range for each node at $250m$ without a fading effect.

In addition, each experiment was run for $4000s$ of simulated time. Our model and system employ the standard DES algorithm for the communication in a community, and the secret keys have a length of 64 bits. This is dependent on the computational capability and characteristics of the nodes within the wireless ad hoc networks. During the simulation experiments described from this point onward, the models and systems

that are used for comparison are all run under identical conditions. We have chosen the following performance metrics for evaluating our community model:

(1) *Hello Overhead*: the number of *Hello* messages broadcast to all neighboring nodes in the network;

(2) *Hello Percentage*: the percent of the number of *Hello* messages broadcast to all neighbors to the total number of traffic messages;

(3) *Session*: a session is defined as the time interval between two consecutive updates of *Hello* messages;

(4) *Management Overhead*: the percentage indicates the sum of all packets sent for key distribution and *Hello* messages out of the total traffic among all nodes;

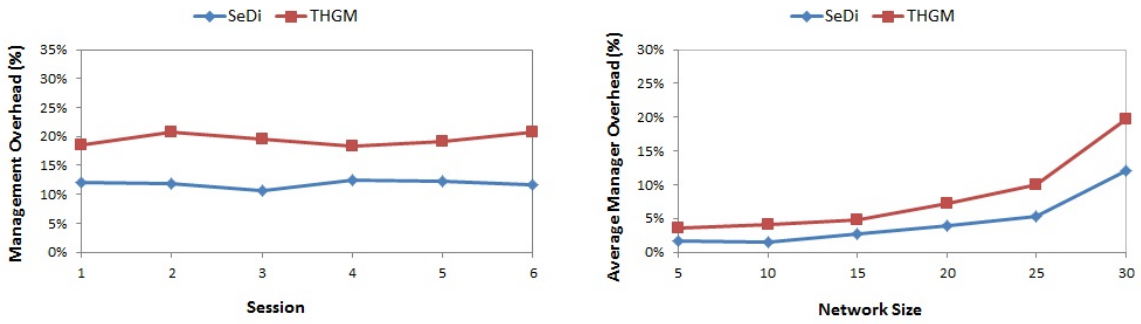
(5) *Network Size*: the total amount of the nodes in the setup network.

We compare SeDi model with the already accepted group management schemes in [11, 58, 61], all of which employ a hierarchical structure to manage the nodes in a network. Thus, we use the two-level hierarchical management scheme and randomly deploy some nodes as the group managers to manage other nodes during the network initialization stage. Let us use the notation n to denote the amount of appointed group managers, and here, $n < N$. These group managers will perform the same operations as the central nodes in their own communities. Thus, this hierarchical model, called the THGM model, can realistically reflect the basic hierarchical group management schemes.

A Comparison between SeDi and THGM Schemes

First of all, we compare the proposed SeDi scheme with the above THGM scheme in terms of management overhead. Figure 3.5 (a) compares the total overhead consumed on management for both models, which include *Hello* overhead, as well as security overhead, etc. We can see that management overhead does not change dramatically on both systems

with the change of the session. We can also note that the management overhead spent on SeDi is lower than that on HGM. This is caused that each group manager in THGM has to communicate more neighbors and even some farther members. Unlike SeDi, in which a node only needs to communicate with its one-hop neighbors, the node in THGM will send more traffic packets in the network if it intends to locate its multi-hop neighbors. Figure 3.5 (b) compares the average management overhead spent on each manager in both SeDi and THGM schemes. Both of them increase the average overhead with more nodes included as the managers, and obviously SeDi has more managers than THGM, so it has a lower average management overhead. Thus, SeDi outperforms the THGM system even though the changes in the value of session follow the same trends.



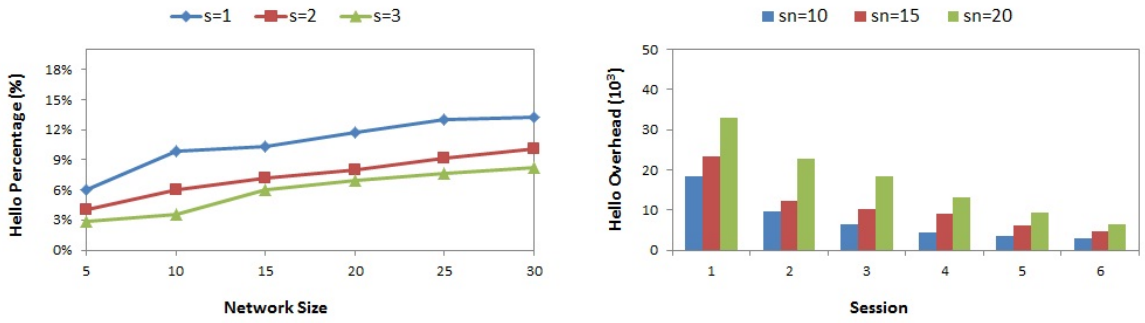
(a) Management Overhead *vs.* Session (b) Average Manager Overhead *vs.* Network Size

Figure 3.5: Average Management Overhead *vs.* Session

The Effects of Community Management on SeDi

Additionally, we study the characteristics of the proposed SeDi scheme, because two main concerns exist: 1) whether the *Hello* message heavily increases the workload of the network; 2) how often the *Hello* message should be updated. Figure 3.6 (a) illustrates that the ratio of *Hello* packets used for community maintenance increases with the increase of network size. Whereas, we can clearly notice that the increase of *Hello* percentage is light even if the network grows greatly. Thus, for a dynamic and wireless environment, *Hello* messages can help a node to determine its neighbors as fresh as possible with

reasonably little cost. Figure 3.6 (b) shows that the overhead spent on *Hello* message decreases as the value of session increases, which means if extending the time interval between two *Hello* messages, *Hello* overhead will be reduced dramatically. Consequently, the broadcast of *Hello* message is necessary and it is also important to find a proper balance between *Hello* overhead and the value of session.



(a) Hello Overhead *vs.* Network Size

(b) Hello Overhead *vs.* Session

Figure 3.6: Hello Message Overhead *vs.* Session

3.4.3 Discussion

The proposed SeDi scheme has many secure and effective attributes to prevent malicious behavior, especially when compared to other conventional group management solutions [6, 11]. For instance, the introduction of a one-hop community eases the complexity of group management; the utilization of both symmetric and asymmetric cryptographic algorithms provides reliable protections for direct interactions in our group prototype, and trust-related technology makes it possible to manage the nodes in a decentralized manner, and thereby improves the groups security during the process of packet forwarding, etc.

As we described in Section 3.3.2, the one-hop community is an independent and local group unit based on each individual node, which represents the most underlying element in our distributed network model. In other words, the number of the nodes in the network determines the amount of the communities. One of its advantages is

that in each community, the central node is only responsible for managing its direct neighbors, from the management point of view. Additionally, our approach has flexible secure group management, owns the smallest distributed group component, and supports the protection of data confidentiality in communications. The trust-based technique in our system achieves membership evaluation and encourages a network's members to contribute to packet forwarding. Moreover, a one-hop community model reduces the global interaction costs and does not need the intermediate nodes to join in the process of communications.

By virtue of introducing the concept of community, it can improve the security of a localized group in a network and thereby enhance the overall security of the network. This is due to the fact that in a network, each community is equally important and the dominated group managers or clusterheads do not exist any more. From the attacker's point of view, each community has the equal management status and each central node shares the same security importance too. On the other hand, in each community, the central node only attempts to manage its one-hop neighbors, thus this central node can directly monitor its members. Supposed that there is any uncooperative behavior caused by its neighbor, the central node can promptly respond and make decision accordingly. At the same time, since a node only communicates with its one-hop neighbors, the total overhead of the global management communication will be distributed to each node, and thereby, the average management overhead of each node decreases accordingly as well. Thus, the distributed community model is different with traditional group management schemes as we discussed in Section 3.2, since it thoroughly employs a distributed approach to manage its neighboring nodes. Therefore, the weakness of group leaders in hierarchical and clustering methods can be overcome and the average cost of management is reduced.

3.5 Summary

In this section, we mainly focus on the issues of secure network management. First of all, we discussed the necessity of secure group management and emphasized its importance especially in wireless ad hoc networks. Then the issues existing in group management were explored along with the features of wireless devices and ad hoc networks. Subsequently, we introduced the concept of community and proposed the SeDi group management model to provide secure and effective network management and group communications. This model aims to ensure the security of each local group in a network and thereby accomplishes protecting the security of the entire network. In SeDi, we employed a combination of symmetric and asymmetric cryptography to achieve secure data communication, and involved the trust-based technique to manage a local community. The analysis and evaluation of these security mechanisms provide the provable secure properties of our management framework, which is resilient to malicious behavior and able to improve the security of a network. Therefore, our SeDi model not only avoids the use of centralized authority, but also simplifies the group management in a fully distributed manner.

Chapter 4

TOMS: A Trust Computational and Management System

As we have already discussed, within open and shared wireless ad hoc environments, some nodes are probably compromised and can become harmful to the network. One direction of current research is exploring how to identify these malicious nodes and thereby exclude them from a presently healthy network. In this section, we introduce the concept of trust and the prototype of trust computing. The trust-based community model can manage nodes dynamically, and the nodes' activities are effectively evaluated in a distributed manner. Furthermore, malicious nodes can be detected based on their trust evaluations, so that they cannot be used in any communication within the community. Thus, determining the trust extent of the nodes is helpful for the improvement of the network's security and reliability.

First of all, we define the concept of trust, discuss the philosophy of trust, and propose a trust-based management system, which includes two important components: a trust computation model (TOMS) and a node evaluation mechanism with assistant trust information (NEAT). TOMS establishes the trust relationship between distributed nodes, calculates a node's trust based on a novel computational model, and specifies a set of rules in this system for wireless nodes. On the other hand, NEAT predicts the

reliability of indirect trust information provided by other nodes. In a wireless network, each wireless node is configured with such a trust-based security system locally, and thereby assesses its neighboring nodes according to their activities and contributions.

4.1 Initiative

In the realm of network security, a new emerging technique that is attracting more and more attentions is *trust* and *reputation*. The traditional notion of trust only indicates a belief or feeling regarding the behavior of peers; whereas, the current notion of trust refers to the outcome of the observations of an expected action, and it changes dynamically [15]. Specifically, *trust* is the credibility of an entity, and *reputation* refers to the estimation or ability recognized by other entities or the public.

In general, an entity will specify how to describe the trust relations established between it and its intended entities. Furthermore, the trust relations are built over the trust evidence gained by the previous interactions of entities within a specific environment. If one entity trusts another entity to perform the intended operation, the trust relationship between these two nodes is considered to be reliable from the interactive initiator's point of view.

If a network is dynamic, uncertain, and agile, the use of a method of trust and reputation evaluation is ideal in evaluating the actions of a node, reducing potential risks to the network's members, and enhancing the overall security of the network. Hence, trust evaluation is gaining popularity as a promising technique to measure the role of wireless nodes and perceive the risks related to the likelihood of encountering malicious behavior. In this way, trust can be viewed as a function of uncertainty. Specifically, if one node believes that *another* node will perform a certain action that it expects, then this node is supposed that it fully trusts *another* node to perform the expected action and there is no uncertainty; in contrast, if this node believes that *another* node will not perform the action as it predicts, the node does not trusts that node, and uncertainty

does not exist either. However, if the node is not sure of whether the aforementioned *another* node will perform the action or not, uncertainty does exist in such a scenario. At this time, the node has to rely on the trust regarding *another* node and makes decision accordingly.

Trust-based systems are introduced in distributed systems or networks, to avoid the central trusted third party swhich is not suitable in distributed and wireless computing environments. Usually, trust is used to record feedbacks about the security evaluations of other nodes. Trust management enables the trust system to track the behavior of each node and make corresponding reactions to the tracked behavior, e.g. rewarding a well-bahved node for good deeds, or penalizing a dishonest node for misbehavior. Thus, effective trust management systems can help well-behaved nodes to avoid working for misbehaving nodes, as well as to detect these malicious ones. On the other hand, the emerging services of trust evaluation aim to ensure distributed and secure management of wireless ad hoc networks, which do not have fixed infrastructures and centralized administration. Malicious behavior can easily tamper such networks, and thereby put ad hoc networks at risk. Additionally, trust evaluations have predetermined characteristics and can decentralize the trust computation throughout wireless networks. Based on the dynamic and flexible nature of the topology, trust management can establish a set of effective rules to make a reliable analysis of certain suspicious nodes based on distributed and incomplete information. Without any doubt, malicious nodes should be detected and excluded from a dynamic and agile network. Therefore, due to the dynamic and distributed characteristics of wireless nodes, the distributed nature of trust is able to deal with these issues well.

Figure 4.1 illustrates a typical example of trust-based wireless communication, in which the importance of trust evaluation is apparent. The notion of "group" is usually applied here, and a node's neighboring nodes are classified as *High*, *Medium* or *Low* depending on their trust levels [16]. In this figure, the sender S establishes a route

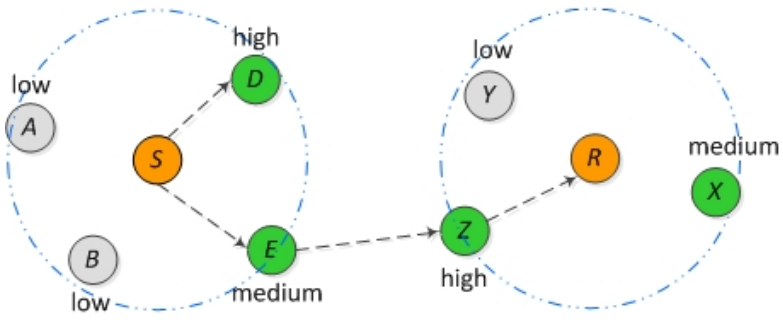


Figure 4.1: The Trust-based Wirless Communication

with a trust requirement "*Medium*" in order to reach the receiver R. As a result, one-hop neighboring nodes E and D can be included in the process of route establishment because their trust levels are higher than or equal to, the required trust level, thereby satisfying the trust requirement. Node Z is also included for the same reason, and so the established route contains only those nodes with a medium trust level or higher. The activities of neighboring nodes can be reported to the sender and the receiver for the purpose of evaluating their reputations. Such trust or reputation evaluations can be shared within the entire network, so that the network can become aware of uncooperative behavior and thus discover selfish or malicious nodes.

In a wireless ad hoc network, a wireless node can obtain new trust credits or lose its trust based on its behavior within a dynamic environment, so only when the node is trustworthy enough for another node can it participate in the communication initiated by that node. Additionally, the security mechanism established between the communicating parties must timely update each node's trust; otherwise, the trust evaluation of the nodes will become stale, and thereby cannot be used for any security purpose. At present, a widely accepted trust evaluation methodology is based on linear function to describe the change of a node in wireless and mobile contexts, because such a linear-based trust evaluation does not greatly increase the overhead of communication in a dynamic and agile network, and is also able to easily handle the flexibility of nodes. In our current research, we mainly make use of the linear-based trust evaluation as our compared model,

which can realistically reflect the fundamental trust computation schemes according to most of these reputation management systems [12, 19, 26].

4.2 Related Work

4.2.1 Trust over the Internet

The earliest concept of trust is presented for the application of Internet security, e.g. E-Commerce (EC) and Multi-Agent Systems (MAS). In Seigneur [96], trust is viewed as the mediating factor of relationships that manages the possible risks. Thus, trust allows the local entity to assess the risk of the proposed collaboration; moreover, the assessment determines the trustworthiness of the other entities so that it can deduce whether the proposed cooperative entities are trustworthy enough to relieve the risk of cooperation. An earlier trust system is Pretty Good Privacy (PGP) [38], in which the web of trust protocol is first described. In PGP, trust signatures can be used to support creation of certificate authorities. A trust signature indicates both that the key belongs to its claimed owner and that the owner of the key is trustworthy to sign other keys at one level below their own. Mondal and Kitsuregawa [78] state that trust builds the bridge between privacy and security and that the trustworthiness of a peer depends fully on the context. Thus, they believe that trust and accountability are interchangeable. Hoffman *et al.* [45] propose that trust can be an approach used to control the information released over the Internet, and thus used as a computing criterion to protect a Web user's private information.

Viega *et al.* [105] describe trust and trustworthiness as the foundation of information security. They discuss how to apply trust in an evolving environment, including the trust assumptions, user input, client application, and execution. Ratnasingham in [92] discusses how the concept of trust can affect the secure management of an organization over the Internet. He emphasizes the importance of trust to the security of the Internet, develops a framework of trust and security, and provides a series of guidelines. Alfarez

and Hailes [4] propose a distributed trust management model and a recommendation protocol as well, in order to judge effectively the trustworthiness of on line users. Buffett *et al.* [20] discuss an issue of privacy over Internet and establish a mathematical model to measure the degree of users are interruptted, which provides partially heuristic theory for our scheme.

PolicyMaker: A General Policy-based Trust Management Framework

Blaze *et al.* [10] investigate the issues of trust managment in X.509 and PGP, and then present their trust management policies using particular security credentials. They propose a comprehensive trust management scheme called PolicyMaker, which specifies the trusted behavior and trust relationships. Thereby, PolicyMaker infers the judgment if an entity satisfies the trust requirement of a third trust party.

In their PolicyMaker trust management system, public keys are bound with policies to describe the actions which it trusts, rather than the identifier of the keyholders. Thus, the security of web services are expressed in an anonymous way without mapping personal identity and authority to credentials and policies. When PolicyMaker processes a request based on a signed message from the holder of a certificate, it will simplify the procedure of traditional authentication services and apply their proposed trust management theories as follows:

- Receiving certificates: 1) verify signatures on certificates and actions, 2) determine the public key of original signers;
- Checking validity of certificates: 1) verify that the received certificates are not revoked;
- Submitting request: 1) bind the request, certificates and description of local policy to a submission message, 2) submit it local trust management engine;
- Proceed if approved.

4.2.2 Trust in Wireless Networks

The concept of trust is also widely applied in ubiquitous and pervasive computing environments. Kagal *et al.* [52] present an architecture based on trust management that is applicable to distributed systems and toward pervasive computing environments. Their security policy is responsible for assigning credentials to entities, delegating trust to third parties, and reasoning about users' access rights. Nekkanti and Lee [80] introduce the trust factor and trust-based level of security, which will make use of different security strategy or cryptographic algorithm according to the variety of trust level. Thus, their proposed protocol can accommodate unauthorized nodes into existing networks. Through their trust schemes, the overhead can be reduced because nodes in different levels of trust will employ different levels of encryption. Zhu *et al.* [124] attempt to establish a secure route from a source node S to a designated node D, and provide an approach to calculate the trust value by applying a delegation graph. The mapping between a delegation edge and an authenticated transitive graph is used to compute the trust value based on the transitive property. Since trust is an crucial part for reputation systems, the dissemination of trust information is helpful for a network. Chen *et al.* [27] propose a trust management scheme for WSNs, in which each node monitors and evaluates the behavior of its neighbors independently, so as to limit the misbehavior of malicious ratings. Two major operations of trust are studied: trust propagation and trust aggregation.

A Trust-related Security-by-Contract Framework

Dragoni *et al.* [36] attempt to solve the problem of pervasive downloading, and provide a layered architecture for pervasive security. They introduce the concept of the notion of security-by-contract ($S \times C$): an application should come with a *contract* which contains the relevant interactions with its host platform; a mobile platform could specify a *policy*, which should indicate its security requirements and be matched by the application's contract. Once a contract is formed, the service provider should be able to negotiate

with the host platform for a specific contract in the following steps: 1) a contract for the mobile application is specified as security requirements and a policy template is used to authorize the contract; 2) once a contract is identified, the contract will be matched with the requirements as defined in the policy to check the compliance of the application; 3) in order to enhance the compliance of contract and policy, some mandatory verification steps take place before the authorized downloading. For exmaple, a contract should come with a trusted signature.

Based on the notion of a mobile contract that a pervasive download carries with itself, trust relationship is established for contract and policy. Their architecture uses a contract to describe the main contents that an application should share with its host platform, and a policy to state the required security specifications. Totally, three layers of application are built as follows: 1) application layer, which provides the administration and certification services in a third party's role; 2) $S \times C$ layer, which has the required service after a contract being authenticated; 3) security layer, which performs the authentication of contract and policy, checks the signature for correctness, and matches the corresponding contract and policy.

Secure Routing based on Trust Evaluation

Yan *et al.* [117] believe that traditional cryptographic solutions cannot fully defend against threats from compromised nodes, so they propose to provide effective security decisions about network activities based on trust evaluations. Each node in the network will process its own logical and computational trust analysis and evaluation. In an ad hoc network, each unit is viewed as a Personal Trusted Bubble (PTB), which includes a wireless device and the ower of the device. Logical and rational trust relationship should be evaluated computationally between bubbles and networks. For their trust evaluation mechanism, many factors are introduced, such as: 1) experience statistics, which logs the data of prior experience accumulated during the communications with other bubbles; 2) data value, which will be higher, if the higher trust needed from other PTBs to transfer;

3) reference which is other bubbles' recommendation, reputation of the evaluated node, etc.; 4) personal preference, which is the bubble owner's personal preference.

Based on their trust evaluation based security solution, trust evaluation is applied to secure ad hoc routing. The protocol mainly consists of route request, trust evaluation, and route reply phases:

(1) The source node broadcasts routing request message to its neighbors, in order to find a route to the desired destination node;

(2) The neighbors of the source node forward the request to their neighbors further if the trust evaluation on the source node passes its predefined threshold;

(3) After getting responses, the source node checks the trust evaluation matrix and conducts the trust evaluation on the responded nodes, to determine which route is preferred (it believes the most trusted);

A Trust-based Clustering and Detection Service

Ngai and Lyu [81] provide a public key authentication service based on a trust model to monitor malicious and colluding nodes in wireless ad hoc networks. Their model allows wireless nodes to monitor and rate each other with an authentication metric. A clustering-based network model is modifided from the *Max-Min* clustering formation algorithm [5] and used to build the architecture of authentication service. In the *Max-Min* clustering formation algorithm, clusters are formed by diffusing the node *ID* and clusterheads are selected based on the higher or lower *ID*. However, the *ID* of a node does not have any special meaning in the protection of the networks security, so trust value is introduced as a criteria in cluster formation: if a node has higher trust value, it is more possible to become a clusterhead. Additionally, a trust model is designed and the trust value can be updated in a linear method, in conjunction with public key certification.

One function of their authentication service is identification and isolation of malicious nodes based on the above clustering-based network model and trust model. In addition

to some straightforward methods to detect malicious nodes (e.g. directly monitoring of individual nodes, or verifying the provided public key certificates), trust values are also used for misbehavior detection. In the exchange of public key certificates, the trust values of the target node are exchanged as well. If the trust value of the target node is lower than a certain threshold, then the target node is judged as dishonest.

A Hierarchical Trust-based Admission Control Management Structure

Virendra and Upadhyaya [106] propose two schemes for trust-based secure information management in wireless networks: one is for central authority-assisted wireless network, and the other is for independent ad hoc network. In the trust-based secure information management system, trust is combined with the use of access control for hierarchical trust establishment and monitoring.

In a central authority network model, there are two servers to act as backbones for the network: admission controller (AC) and global monitor (GM). The role of AC is to allow or disallow a new wireless node to enter the network. The GM handles all other network health management issues and ensures the appropriate functionality of the network. When a new node intends to join the network, the AC makes admission decisions depending on the trust value of the node and its intention. Moreover, the trust of all nodes is classified to three levels: *Low/Unknown*, *Medium* and *High*. For an individual node, AC/GM allows a node to implement a certain operations based on its intention and generates an intention map for those operations. When managing the trust in the network, a hierarchical trust model is used to address scalability and also to reduce control overhead on the head nodes. A 3-tier structure is formed by the underlying trust domains, domain heads and a network head. Thus, the trust-based information management system defines a trust based admission control scheme based on graph theory to update changes in trust and employs the domain head to achieve trust management.

4.2.3 Price-based Schemes

Another alternative to the trust- and reputation-based strategy is the pricing- and payment-based mechanism. Liu and Krishnamachari [68] attempt to establish a stable and reliable routing path that involves the pricing mechanism, in which the destined node pays for each successfully delivered packet. Zhong *et al.* [121] design a credit-based system named Sprite, which provides an incentive to selfish nodes and requires that each node periodically reports its recent activities to a Credit Clearance Service (CCS). Ileri *et al.* [48] propose a pricing mechanism that sets the price for the occupation of channels and for the action of forwarding packets.

Al-Karaki and Kamal [3] compare the advantages of rewarding strategies and punishing strategies. Moreover, they point out that the punishment-based strategies may be preferred, because selfish nodes perhaps do not care about rewards; however, these nodes are more interested whether they can access resources and continue to utilize the network. Consequently, punishment strategies can stimulate selfish nodes to take part in cooperation more effectively. A truthful multipath routing protocol (TMRP) is proposed in [112], which takes advantage of an auction mechanism for providing incentives to packet forwarding. The auction-based approach creates as little as possible overhead when choosing the next hop during the procedure of packet forwarding. A market-based approach in [73] quantifies the benefits and costs of forwarding packets by setting prices for data forwarding. For example, intermediate nodes will get rewards if they provide resources, or source nodes will also be charged for sending packets because they exploit others' resources. This approach forces the source nodes to avoid unnecessary data transmission, so a utility function is introduced to calculate the benefit of transmitting a packet.

4.2.4 Linear-based Trust Model

In most reputation evaluation models, the reputation value is generally computed based on the linear function, because it can adapt to the flexible characteristics of wireless devices but does not cost a number of computational resource of an ad hoc network. Though there are other trust evaluation mechanisms applied in the field of network security [4, 105], they are not well suitable to wireless and ad hoc environments, due to the reasons that they ignore the issues of nodes' mobility and limited computational capability in the wireless network. In particular, some of them need a heavy bootstrap process and/or complicate the computation of trust, so that overmuch computational overhead is consumed on reputation computation. Thus, the linear-based mechanism is the most basic methodology, but it is also the most widely accepted one. In our current research, we mainly make use of the linear-based trust evaluation as our compared approach, which can realistically reflect the fundamental trust computation schemes according to most of these reputation management systems [27, 81, 117].

4.2.5 Trust-based Evaluation

In wireless networks, nodes have to rely on the cooperation of other nodes, and forwarding data for each other will incur costs for a node (e.g. energy, channel occupation, etc.). Therefore, the reasonable encouragement of cooperation can result in significant improvement in a network's performance. As shown in Table 4.1, we compare some existing trust-based schemes based on the characteristics of trust calculation and evaluation.

Node Evaluation in Wireless Networks

The trust-based technique is widely applied to the realm of node evaluation. Balakrishnan *et al.* [7] propose a trust model to address the concerns arising from reputation recommendations. Their model helps determine, in the decision making phase, whether to accept a packet forwarding route or not, depending on the evaluated trust of all nodes

along the route: the combination of direct and recommended trust. Ghosh *et al.* [39] emphasize the importance of honesty from a node's opinion on trustworthiness recommendations by introducing the collaboration of nodes. Singh and Liu [97] present a secure and anonymous protocol (TrustMe) for trust management, which provides anonymity protection for trusted hosts. In particular, it allows a user to issue a query message for a trust value; and at the same time, the user can obtain the true trust value in the presence of malicious users.

Table 4.1: The Comparison of Trust Computation and Evaluation Schemes

Algorithms Characteristics	**Linear -based**	**Credit or Price-based**	**Historical -based**	**Direct Trust**	**Indirect Trust**
Hoffman [45]	linear	-	-	direct	-
Viega [105]	-	-	-	direct	indirect
Al-Karaki [3]	linear	price-based	-	direct	-
Zhong [121]	linear	credit-based	historical	direct	-
Blaze [10]	linear	-	historical	-	indirect
Chen [27]	linear	-	-	direct	indirect
Ileri [48]	linear	price-based	-	direct	-
Dragoni [36]	linear	-	-	-	indirect
Yan [117]	linear	-	historical	direct	indirect
Ngai [81]	linear	-	historical	-	indirect
Wang [112]	linear	auction-based	-	direct	-
Marbach [73]	linear	price-based	-	direct	-
Cohen [26]	-	-	historical	direct	indirect
Sun [100]	linear	-	historical	direct	indirect
Ding [35]	linear	-	-	direct	indirect
Buchegger [19]	linear	credit-based	historical	-	indirect

(1) An Information Theoretic Framework for Quantitative Trust Measurement and Propagation

Sun *et al.* [100] design a distributed information theoretic framework for trustworthiness evaluation in ad hoc networks, to compute, maintain and update each node's

trust record according to the node's behavior in terms of packet forwarding. First, they propose some trust metrics to discuss the characteristics of trust,

- Trust is unpredictable: a nod's trust to another represents the probability that the node's agent will perform the expected action;
- The propagation of trust is not concatenated: with the inclusion of trust recommendation, the trust information cannot be increased through propagation;
- Multipath propagation of trust does not reduce trust: in the case where a node receives different recommendations for the agents from multiple sources, multipath recommendations will not increase uncertainty;
- Independent trust recommendation is the threshold of correlated recommendations: when the trust relationship is established jointly through concatenation and multipath trust propagation, those correlated recommendations should not be higher than the trust built upon recommendations from independent sources.

Subsequently, the authors present two trust measurement models,

(1) Entropy-based trust model: in this model, a node's trust value is an increasing function with the probability p represented by trust. The trust propagations are calculated directly from trust values defined in *H(p)*, in which the *initiator* trusts the agent the most, and the trust value is 1, when $p = 1$; the *initiator* distrusts the agent the most, and the trust value is -1, when $p = 0$; when $p = 0$, the *initiator* has no trust in the agent and the trust value is 0.

(2) Probability-based model: in the second model, they calculate concatenation and multipath trust propagation using the probability values of the trust relationship. Then, the probability values can be easily transferred back to trust values using the first model (entropy-based trust mode).

As an important component, their scheme also makes reputation recommendations to other relevant nodes, to assist malicious node detection and route selection. When the *initiator* obtains the trust value from its observation, it respectively uses probablistic approach and Bayesian approach to estimate of the probability led by the trust value. For example, in the first probablistic approach, an unknown parameter θ, is the probability of performing the action at each trial, it is estimated given the fact that k actions have been performed out of N trials. Assume that node A would like to ask node C to transmit packets, while A does not have trust relationship with node C. Since A checks with node B for trust recommendations, B has made recommendations to A for N times. After node A receives the recommendation from node B, A calculates the recommendation trust of based on B's previous recommendations using either probablistic or Bayesian approach. Eventually, A calculates the trust in C about packet forwarding through the concatenation propagation.

(2) CONFIDANT (Cooperation Of Nodes - Fairness In Dynamic Ad hoc NeTworks)

Buchegger [19] proposes a protocol named CONFIDANT which includes a reputation system to cope with falsified trust information. Individual node is able to to detect misbehavior by first-hand observation and use of second-hand information provided by other nodes. A reputation system is introduced to make use of all the information available. Such a fully distributed reputation system that can cope with false information and effectively use second-hand information in a safe way. In particular, the proposed reputation system makes use of machine learning estimation and classification techniques for filtering observed information, in which the rating of a node's reputation can reflect the quality of its behavior. In their approach, each node maintains a reputation rating and a trust rating about all other nodes it cares about. Reputation ratings capture the quality of the behavior of a node as an actor in the network performing routing and forwarding requests. Based on a modified Bayesian estimator using the *Beta* function,

second-hand reputation information is only accepted if it complies with the current reputation rating. Trust ratings capture the quality of a node as an actor in the reputation system and reflect whether the reported first hand information summaries published by node are likely to be true. These trust ratings are updated based on the compatibility of second-hand reputation information with prior reputation ratings. The viewpoint that a node has about the behavior of another node is captured in a reputation system, which is used to classify nodes as misbehaving or normal. Once a misbehaving node is detected, it is isolated from the network.

Node Evaluation in Vehicular Networks

A vehicular ad hoc network is a special form of wireless ad hoc network that provides communications from vehicle to vehicle (V2V) and vehicle to RSU. The concepts of trust and reputation are also widely applied here to stimulate cooperation from most vehicles in the network. Park *et al.* [83] propose a reputation system for VANET based on two scenarios of RSU: one is the RSU with Internet access, and the other is an isolated RSU without Internet. Whichever RSU is communicated with will issue a reputation message with the corresponding reputation certificate to a vehicle in its region, keep monitoring this vehicle's daily behavior, and thereby update the vehicle's reputation accordingly. Wang and Chigan [113] design a trust-token scheme for cooperation enhancement among vehicular nodes. In their mechanism, symmetric and asymmetric encryptions are adopted to protect packet integrity, and neighborhood watchdog is used to generate trust token, so that the cooperation among vehicular nodes during packet disseminations may be stimulated.

(1) A Trust Opinion Aggregation Scheme

Chen *et al.* [26] establish a trust opinion aggregation model to combine the identity-based aggregate signatures from multiple messages into one aggregate signature. In a vehicular ad hoc network, the aggregation of several signed messages merges the different

trust opinions and eliminates the duplicate trust opinions. During the process of trust information aggregation, a lot of secondary trust information are accumulated and third-party nodes take part in the aggregation of trust opinion. Totally, three steps consist of the identity-based aggregate signature algorithm: 1) Message and trust opinion signing; 2) Trust opinion aggregation; 3) Signature verification. The concept of "Bilinear Maps" based on identity-based encryption is introduced in these three steps.

(1) Trust opinion signing: two types of identities will sign messages: the sender, which is the message originator, and the evaluator, which gives a trust opinion.

(2) Trust opinion aggregation: a third party can combine the original signed message with n signed trust opinions from n distinct nodes, or combine two aggregates on the same message M into a larger aggregate, and there may exist duplicate trust opinions.

(3) Signature verification: the verification of a message is mainly based on two different cases, whether a signed message has trust opinion or not.

(2) An Event-based Reputation Model

Ding *et al.* [35] propose a reputation system to filter false warning messages based on event reporting. In a VANET, vehicles are allocated a variety of roles: event reporter (*ER*), event observer (*EO*), and event participant (*EP*). Figure 4.2 shows the role assignment of different vehicles in a VANET when some events occur.

- Event Reporter (*ER*): a vehicle can perceive incidents by equipped sensors, and then send alarm messages to other neighboring vehicles (bogus warning messages may be issued by malicious *ER*)

- Event Observer (*EO*): within one hop of an event reporter, vehicles can observe the behavior of event reporter when they receive the event message from it.

- Event Participant (*EP*): other vehicles beyond one hop of an event reporter are regarded as event participants, since they can receive and forward the event message, but it is impossible to identify the behavior of event reporter.

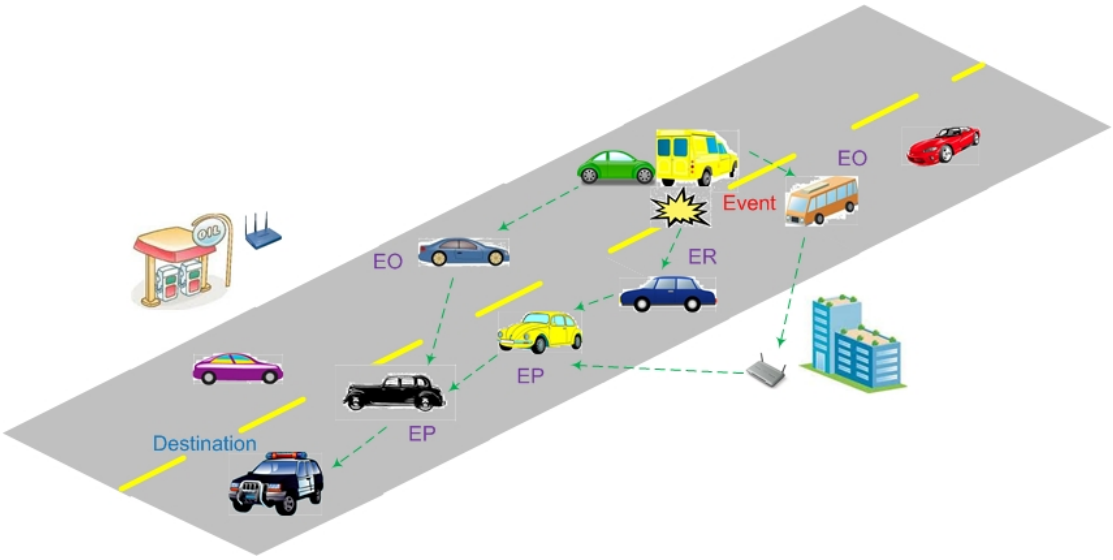


Figure 4.2: Event Reporting in a VANET

When an *ER* encounters a traffic event, its equipped sensor will collect traffic-related data. Depending on the detection frequency, *ER* can estimate the severity of this traffic event and the event message will be sent to all neighboring vehicles (*EO*) in one hop. In the process of reputation calculation, *EO* will be a very important role to identify bogus event messages. When an *EO* receives a traffic warning message from an *ER*, it first stores this message into the event table if there is no duplicate message record in the table. In order to prevent an *ER* from sending bogus messages, event observer will observe the succeeding behavior of *ER* to estimate the reliability of the received event report message, though it does not observe the reported event directly. A typical behavior model is applied here to match the behavior of *ER* and measure the trustworthiness of *ER*'s reputation. Intuitively, if the behavior of *ER* matches the behavioral model related to the traffic event type, the event message is considered as trustworthy. Nevertheless, if

an *EO* found other behavior of an *ER* deviated from standard behavioral model, it can judge that this event message from the *ER* may be bogus. Accordingly, the reputation value of this event message is lowered by this corresponding *EO*.

Outlier Detection

In ad hoc scenarios, misbehavior of nodes due to selfish or malicious reasons is fatal to the performance of wireless or vehicle ad hoc networks. In order to detect abnormal behavior, the evaluation of a user can help locate misbehavior in a network. Wang *et al.* [111] study the threats from selfish and malicious nodes respectively. Selfish nodes generally do not intentionally damage other nodes' service, though they refuse to provide their resources to others. Unlike selfish nodes, malicious nodes always deliberately initiate a variety of attacks that jeopardize the functions of a network: thus, the prevention of this kind of node is apparently more important. In one of the earliest works on the observation of neighbor's behavior, Marti *et al.* [74] discuss the issue of misbehavior detection in a MANET. Their proposed algorithm first classifies all nodes based on their measurable behavior. Subsequently, the algorithm uses a watchdog mechanism to detect malicious nodes and thereby provides a detour from these detected malicious nodes by using a pathrater mechanism.

Li *et al.* [63] develop a collaborative outlier detection algorithm, by which each node in the wireless network is able to evaluate its intermediate neighbors locally and exchange evaluations with these intermediate neighbors until there is no further update to change. In [62], they also provide an outlier detection algorithm which employs neighboring nodes' observation to prevent misbehavior like packet dropping, and etc. In [57], misbehavior detection relies on an event reliability factor (ERF) in a WSN. When an event is found by a certain sensor, it is difficult for the individual sensor to judge if the monitored event is true or not. Thus, with nodes surrounding the event, information is exchanged which aims to calculate a truthful ERF. Consequently, the true event is sent by the monitoring node and the false event is blocked in the local community.

4.3 Philosophy of Trust

Trust-based systems are introduced to avoid the central trusted third authority which is not applicable in pervasive and wireless computing environments. With the notion of trust, reputation is used to indicate the belief about how to deal with a situation with uncertainty. Here, we first describe the specifications of trust and present the definition of trust relationship.

4.3.1 Trust

Traditionally, trust has only been used as a belief or feeling regarding the behavior of peers [53]. The definition of trust can be traced back to the social science literature, which have been interpreted in different meanings: reputation, expected recommendation, probability, and so on. Actutally, trust is a relationship between two entities such that one entity believes, expects, and accepts that the other trusted entity will act or intend to act beneficially [77]. In our research, we define the concept of trust as follows:

"*trust represents the degree to which a node would be trustworthy, secure, or reliable in any interaction with the node*".

The extent of the trust is measured by a continuous real number and we denote the trust value as T. In some cases, trust and reputation are interchangeable, because accumulative trust can reflect the extent of reputation. Thus, trust and reputation are used to evaluate other nodes' abilities to fulfill an expected action, and a node can take advantage of this trust-based information to make decisions when it must choose a set of trustworthy nodes for its packet forwarding. During the nodes' communications, only when a node is trustworthy enough for another node and satisfies the trust requirement, can it participate in the communication initiated by that node. Thus, the trust relationship should be cnosidered as unidirectional. To that end, a node can also have different trust values when it is evaluated by different nodes.

Since wireless ad hoc networks do not utilize any infrastructure for data transmis-

sion, their packet forwarding is accomplished via cooperation between neighboring nodes. Therefore, based on this special need for trust and reputation evaluation, the specification and computation of trust is obviously significant for dynamic and agile wireless networks.

4.3.2 Trust Relationship

The relationship between a pair of entities can be unilateral or bilateral, which is also applied to trust. The notation $R_{(i,j)}$ indicates the relationship between two entities i and j. Let us denote the two entities as *subject* and *object* and the relationship between the *subject* and *object* can be described in the following way:

- The relationship of *object* to *subject* is $R_{(subject,object)}$;
- The relationship of *subject* to *object* is $R_{(object,subject)}$.

When it comes to trust, the relationship is more specific. First, we designate the trust relationship between two entities is established over trustworthy nodes: one node, called the *trustee*, which can forward packets for another node, called the *trustor*, which is the initiator of communications. Here, we take advantage of packet forwarding as a typical example for the expected action or behavior between the nodes, due to the features of a wireless ad hoc network. If and only if the node *trustor* trusts the other node *trustee* to forward packets, the *trustor* will send its packet to the *trustee* for packet forwarding; otherwise, it will not initiate the request of packet forwarding. Then, we specify that trust is a unilateral relationship established between these two nodes for a specific purpose or action. Similarly, $T_{(i,j)}$ represents the trust relationship between node i and node j. Thus, the notation $T_{(trustor,trustee)}$ indicates the unilateral relationship established between node *trustor* and node *trustee*. At the same time, $T_{(trustor,trustee)}$ means that the node *trustor* would request the node *trustee* to forward packets for it, when *trustee* meets its trust requirement. Let us assume the communication between nodes A and B, in which A is the *trustor* and B is the *trustee*, then

- The trust of B to A is $T_{(A,B)}$;
- The trust of A to B is $T_{(B,A)}$.

In general, if the nodes' behavior has been faithful to the reputation evaluation system, then trust will increase between these entities. For instance, as the initiator of an operation, A first sets a trust value Tr_A as its trust requirement. If the trust of B satisfies the trust requirement ($T_{(A,B)} > Tr_A$), A allows B to forward packets for it. Moreover, if B successfully forwards a packet for A, then B is considered to be an honest node for A and A thus increases its trust value $T_{(A,B)}$ for B's good behavior; otherwise, if B lies about or exaggerates its contribution to routing, then B is a suspicious node that will be penalized and $T_{(A,B)}$ decreases accordingly. We call the number of a node's activity as the node's trust evidence, and it will increase once it successfully cooperates with the communicating initiator and *vice versa*.

Assume that the *trustor* node trusts the *trustee* node to perform the intended operation, the trust relationship between these two nodes can be established reliably from the communicating initiator's point of view. In other words, $T_{(i,j)}$ is not necessary to be symmetric. That is, if a node i trusts another node j, i still can communicate with j or let j forward packet for it, even if the node j may not trust i. However, if the node j does not trust node i, j will not allow i forward packet for itself, but j may still forward packets for i, as j is an honest node for i. Figure 4.3 illustrates the scheme of communication that takes into a node's trustworthiness into account and updates the node's trust value according to its behavior, in which S allows two neighbors with satisfactory trust levels, D and E, to forward packets for it. In the meantime, these trust values $T_{(S,E)}$ and $T_{(E,Z)}$ both increase for their successful forwarding. As a result, the feedback from the *trustor* regarding the behavior of the intermediate node is important to the evaluation of the *trustee* and the security of the network. Trust is clearly an effective recommendation mechanism that will greatly enhance the security of the network, and indeed, provide the reliability necessary for peer-to-peer exchanges of information.

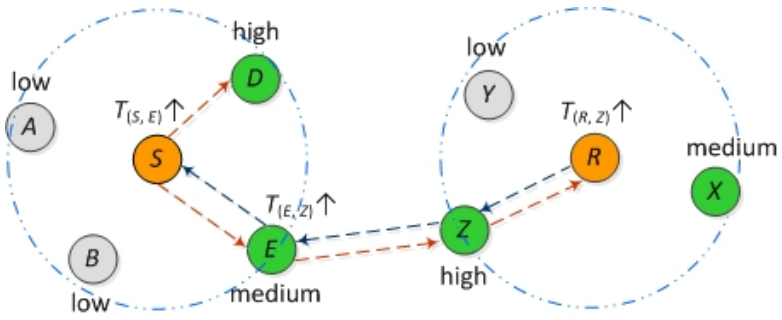


Figure 4.3: The Schematic Diagram of Trust Evaluation

4.3.3 Trust-based Community

The definition and evaluation of trust affect not only the specification of security mechanisms, but also the management of a network. Based on the above descriptions of trust and trust relationship, we can apply the method of trust management to our community management. As we described in Section 3, a node that is a central node will include its one-hop neighbors into its community, and in such a community, the *trustor* node is the central node, and the potential *trustee* nodes are one-hop neighbors of the central node. Meanwhile, the trust relationships play an important role when establishing a trusted community. These relationships involve a devoted relation between two nodes, in particular to represent that their behavior is trustworthy and reliable to each other. In this way, the notation $R_{(Subject,Object)}$ can be expressed as $T_{(C,Mem)}$ to indicate the trusted relationship established between these two nodes: the central node C and its one-hop member Mem. If the central node C trusts its neighbor Mem to perform the intended operation, the established trust relationship between these two nodes is deemed to be reliable from the communicating initiator's point of view.

As such, the concept of trust can be used by the central node to evaluate other members' ability to execute an expected action, and thereby the central node is able to employ the trust-based information to make the corresponding decisions. Let us assume a similar communication scenario to Section 4.3.2 between the central node C and one of

its neighboring member *Mem*, in which *Mem* has already met the trust requirement of *C*. If the member *Mem* performs some good deeds for *C*, then *Mem* is considered to be a cooperative node for *C* and *C* thus increases its trust value $T_{(C,Mem)]}$; otherwise, if *Mem* refuses to contribute to *C*'s routing or traffic relay, then *Mem* is deemed a uncooperative node that its trust $T_{(C,Mem)}$ will be lessened accordingly.

4.4 A Computational Trust Model

Given the definition of trust and trust-based relationship, it is significant to find an effective method to calculate a node's trust. In most trust computation models, the trust value is computed generally based on the linear function. However, in this section, we propose a novel trust model that will update the trust value based on different increase-shapes. In this way, our computational model can improve the flexibility and reliability of trust management.

4.4.1 Initial Consideration

First, we will investigate several different types of mathematical functions and observe their shapes. We then explain how our trust scheme is deduced, based on these mathematical functions.

Exponential Functions

Exponential functions are functions of the form $f(x) = a^x$ for a fixed base a which could be any positive real number. Exponential functions are characterized by the fact that their growth rate is proportional to their value. Though the shape of the graph $y = a^x$ depends on whether $a < 1$, $a = 1$, or $a > 1$, we only use $a > 1$ to describe a node's trust increase in our scheme. Thus, the exponential function $y = a^x$ $(a > 1)$ has a slow increase shape when x is not a large number (for example, $x < 1$), and y will increase slowly with the increase of x. Such functions are suitable for measuring the nodes with

low contribution of packet forwarding or significant uncooperative behavior. Figure 4.4 illustrates a few of exponential functions such as $y = 2^x$, $y = 2.5^x$, and $y = 3^x$.

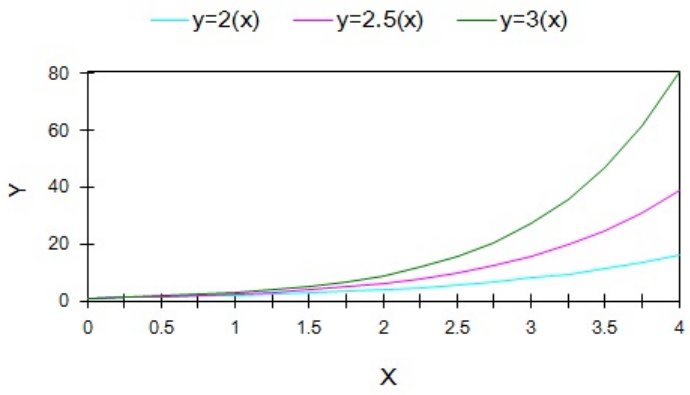


Figure 4.4: Graphs of Exponential Functions

Logarithmic Functions

The logarithmic function is sometime defined as the inverse function of the exponential function, and it has the form $f(x) = \log_a^x$. As we know, the graphs of logarithmic functions and exponential functions are symmetrical with respect to the straight line $y = x$. Figure 4.5 shows that logarithmic functions $y = log_2^x$, $y = log_{2.5}^x$, and $y = log_3^x$ increase quickly with the increase of x, when x is not a large number (for example, $x < 1$). Therefore, logarithmic functions have a fast increase shape when compared with exponential functions. In our trust evaluation scheme, logarithmic functions are used to measure the nodes with a large number of contribution to packet forwarding or little uncooperative behavior.

Linear Functions

Finally, we discuss the simple linear functions, which generally they have the form of $f(x) = ax + b$. Here, we only consider the simplest linear functions whose graphs pass

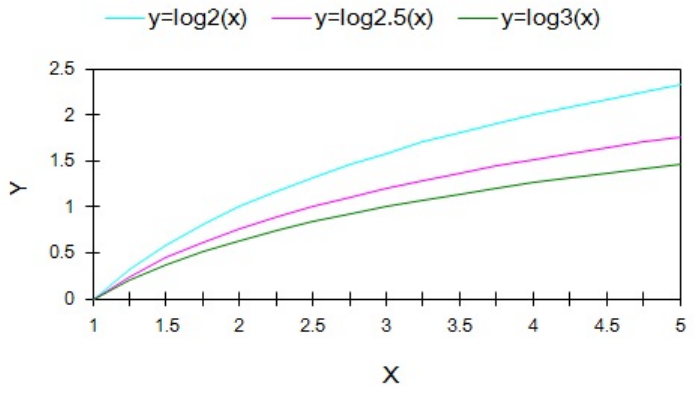


Figure 4.5: Graphs of Logarithmic Functions

through the point $(0, 0)$; these functions have the form $f(x) = ax$. Since linear functions have a stable increase shape, they are used to measure the nodes with a stable change in trust or constantly cooperative behavior. As shown in Figure 4.6, linear functions $y = x$, $y = 0.5x$, and $y = 1.5x$ increase moderately with the increase of x. Thus, we conclude that linear functions have a medium increase shape when compared to logarithmic functions and exponential functions.

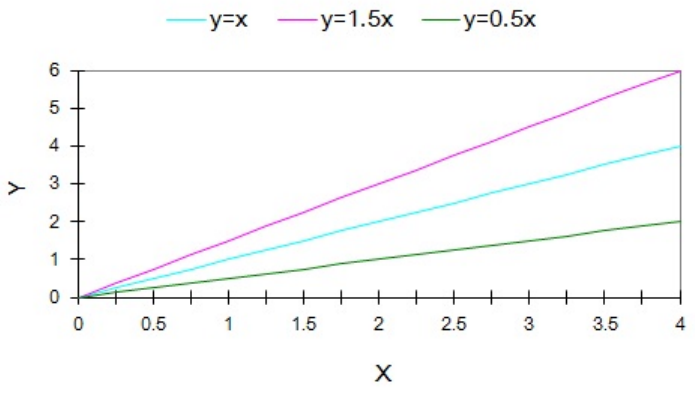


Figure 4.6: Graphs of Linear Functions

4.4.2 Model Overview

Our Proposed Trust Computation Theory

Based on the above discussion and observations of different mathematical functions, we summarize the following heuristic outcomes:

- Logarithmic functions have faster increase shape and will be used for cooperative nodes;
- Exponential functions have slower increase shape and will be used for uncooperative nodes;
- Linear function have moderate increase shape and will also be used for those nodes which have medium contribution for packet forwarding.

Furthermore, we propose the theory that our trust evaluation scheme builds on:

- A node is only allowed to participate in the communication initiated by the source node when this node is trustworthy enough for the source node.
- A cooperative node will be rewarded for honest behavior, such as successfully forwarding; an uncooperative node will be penalized for malicious behavior, such as packet dropping.
- A node's past historical trust records are introduced as a significant factor in order to measure its current trustworthiness.
- The principle of trust evaluation will reward nodes with good past trust records more, nodes with bad past trust records will be rewarded less, and nodes with medium past trust records will be moderately rewarded.

Modeling the Trust

Here, we propose a novel trust-based reputation prototype that will update the trust value based on different increase-shapes, so that our model exhibits adaptive and intelligent characteristics. This proposed reputation model is greatly different with those traditional schemes we described in Section 4.2, since it takes advantage of various mathematical methods to achieve the effective trust computation, and simplifies the complexity of trust evaluation.

In our TOMS system, a node's recent trust which is defined as the node's historical trust records and the past behavior of n_i, has a significant effect on its current trust evaluation, so the recent trust of the node n_i is denoted as rt. For the trust metric, two factors are taken into account: the residential time T_m and the recent activity ra. When a node n_i stays in another node's community, the residential time of the node indicates the extent of its trustworthiness, since the longer T_m is, the longer n_i stays in the community and thus the more trustworthy n_i is. Otherwise, a malicious node would be removed from the community and could not survive for a long time. Specifically, the time T_m is an accumulative value, measured in a time unit such as ms. Thus, the time T_m that the node stays in the community is one factor of the trust metric. The other factor of the trust metric is the past activity record of the node. That is, the recent activity ra of the node n_i can log the past behavior and reflect the reputation of n_i. A counter will be used to record ra, which records the amount of the node's past activities. As a result, we specify the trust as a function that depends on the time that a node has spent in the community, and on the past trust which this node has had in recent periods [20]. First, we define

$$\begin{cases} 0 < rt < 0.5 & 1 < \mu < 2 \\ rt \approx 0.5 & \mu \approx 2 \\ 0.5 < rt < 1 & 2 < \alpha < 3 \end{cases} \quad \Longrightarrow \quad N = \mu \times rt \tag{4.1}$$

here, μ is an amplifying factor, and N is a decimal to differentiate all nodes through their recent trust rt. Both N and μ depend on rt, and they are determined by the individual node. Through the factor μ, N will yield a value very close to 1 for nodes with a moderate trust ($rt = 0.5$), a value below 1 for nodes that have lower trust ($rt < 0.5$), and a value above 1 for nodes that have a higher trust ($rt > 0.5$). Then, let ω denote the other factor

$$\omega = \kappa^{T_m} \times ra \tag{4.2}$$

where κ is a discount factor between 0 and 1 and ra is the node's recent activities, which can include a successful forwarding or a deliberate exaggeration. Finally, the trust metric is evaluated as follows:

$$\begin{cases} T = \dfrac{\lambda \times (1 - N^{1+\omega})}{1 - N} & N \leq 1 \\ T = \dfrac{\lambda \times (N^{1+\omega} - 1)}{N - 1} & N > 1 \end{cases} \tag{4.3}$$

where λ is a scaling factor to keep the trust value T at a value between 0 and 1. Although each node independently selects the values for μ, κ and λ , the factors will keep the node's trust value in the same comparable range. For example, κ and λ may be some values between 0 and 1. Accordingly, the increase in trust will have three shapes depending on the past trust value and the time that the node has stayed in the community. If the node n_i has had a good trust record in the past, then its current trust will increase quickly; if n_i has fewer trust credits, its trust will increase slowly; finally, for a node n_i with a medium trust record, its trust will increase moderately as well. Hence, the TOMS model can distinguish nodes based on their behavioral history, from an intelligent point of view.

Based on the maintenance of the community, the central node will update the trust value each time, based on the periodically broadcast *Hello* messages. We define the time interval between two consecutive updates of *Hello* messages as a session s. At the end of each session, the central node clears the variables T_m and ra respectively, and uses each node's current trust value T to replace its corresponding recent trust rt. If the

trust value is not updated for a long time, the trust value may be stale or out-of-date, so that it cannot reflect the realistic activities of the node. At the end of a *Hello*-based session, it does not nullify the reputation obtained so far, instead it just transmits the current trust value to the recent trust value. Thus, the latest evaluated current trust value will be used as the foundation of trust evaluation in next session. The function of *Hello* message is to help determining the neighbors of the central node in a community. In particular, such *Hello* messages are only sent at the beginning of each session and will not be broadcast during other occasions.

4.5 NEAT: A Node Evaluation Scheme with Assistant Trust

For a wireless ad hoc network, in which nodes are exposed to an open and agile environment and node cooperation plays an important role, misbehavior is a primary threat for the cooperation between nodes. As an emerging technique, trust gains extensive attention as it is deemed as an effective and practical distributed management method to perceive abnormal behavior. However, malicious or selfish nodes do not always comply with the specifications of network protocols, in order to greedily utilize the resources of other honest nodes. If the behavior of each node in a network can be evaluated timely and accurately, this will be significant to the prevention of misbehavior or uncooperative behavior. In this section, we explore the challenges of misbehavior detection, and emphasize the importance of node cooperation. Based on the notion of trust evaluation, we then propose a set of novel misbehavior mitigation solutions with the assistance of trustworthy neighboring nodes. Thereby, an outlier detection scheme is presented based on Naïve Bayes (NB) algorithm, which is used to measure the reliability of trust information provided by other adjacent nodes. Thus, our schemes can evaluate a nodes' trustworthiness properly, and mitigate the negative effects caused by misbehavior accordingly. We examine the scheme based on our risk assessment criteria and attack models. Through

the security analysis, Naïve Bayes makes the trust-based outlier detection more suitable and reliable for distributed wireless ad hoc networks.

4.5.1 Preliminaries

With the increasing prevalence of wireless devices, the use of a distributed method is in accord with the secure requirements and vulnerable features of wireless devices. As described above, trust is introduced to measure the trustworthiness of a node in such a context, which is intended to participate in an expected operation, such as packet forwarding. Apparently, the evaluation of a nodes trust can reduce potential risks to a network's member, by detecting abnormal behavior, and thereby enhancing the security of the overall network. During the process of trust information collection, it is important to measure the reliability of the collected trust information from other nodes. Malicious nodes may threaten the security of the network if they inject incorrect trust information, and thus disturb the course of trust evaluation. As a result, some strategies are necessary to provide protection for trust evaluation when this process needs auxiliary information from other sources.

Here, two major avenues of trust-related research are drawing a great deal of interest: (i) how to measure the reliability of collected trust information, and (ii) how to mitigate the effect caused by misbehavior related to trust evaluation. As we described in Section 4.4, we have already developed a trust-based computational model; this encourages nodes in a wireless ad hoc network to contribute to network cooperation, preventing node misbehavior: packet dropping (*black hole* attack). However, we here concentrate on the methods of evaluating a node's trust and predicting the reliability of provided trust by other nodes, in order to prevent those trust-oriented attacks which aim to disturb the procedure of trust evaluation and to affect the rational result of trust evaluation; this is because we desire to provide a secure process of trust evaluation to guarantee that the incorrect trust information is involved as little as possible when evaluating a node's trust.

Therefore, secure node evaluation can monitor a node's behavior reliably, and thereby improve the security of the entire network.

We propose a node evaluation scheme with assistant trust, which is referred to as NEAT. In the design of NEAT, a certain amount of neighboring nodes are appointed as assistants to provide auxiliary trust information, to evaluate a node justifiably. As methods of evaluation, both un-weighted and weighted trust evaluation are employed to mitigate the effect of misbehavior in the network. On the other hand, a supervised learning scheme based on Naïve Bayes is designed to judge the reliability of provided information in relation to trust, and to validate the collected trust data. In our scheme, as a node initiates the evaluation of trust to another node, it will query other nodes concerning the evaluated node's trust. After obtaining feedback from other appointed trust assistants, the node adopts the Naïve Bayes algorithm to predict the reliability of those nodes which provided trust values, before they are used for trust evaluation. Thus, the Naïve Bayes classifier can detect the outlier of trust values and filter false or unrealistic trust information effectively, so that our outlier detection scheme evaluates the nodes in a reliable way, which is suitable to dynamic and agile environments.

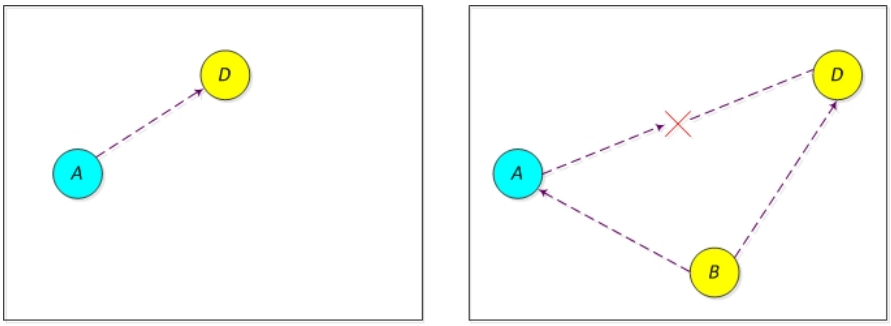


Figure 4.7: Illustrations of Direct and Indirect Trust

4.5.2 Fundamental Concepts

Direct Trust and Indirect Trust

Borrowed from concepts in social science, trust is divided into direct trust and indirect trust [4]. Direct trust stems from the first-hand observations locally obtained by the host (central node), while indirect trust refers to the recommendations released by third parties (other nodes). Figure 4.7 illustrates the schematic diagrams of direct trust and indirect trust. In a wireless ad hoc network, direct trust cannot always provide comprehensive evaluation of the target node due to exterior circumstances, such as channel conditions, temporary unavailability, interference, and etc. At this time, indirect trust is used to provide secondary information to help evaluate the target node's actual trust. In our scheme, we combine direct trust and indirect trust to evaluate a node's eventual trust, so as to reduce the cooperative opportunities of malicious nodes.

Node and Role Definition

For the roles of a node in the procedure of node evaluation, we classify all participating nodes in three categories:

- *Evaluating node*: in the process of node evaluation, the evaluating node is the central node in a community and will initiate the trust query to other nodes;
- *Queried node*: the queried nodes are those nodes which receive the trust query from the evaluating node. Specifically, these nodes are the trust assistants in the community;
- *Evaluated node*: it is the target node which is evaluated by the evaluating node, and its trust information is gathered from the queried nodes.

Naïve Bayes Algorithm

First, we explain the theory of Naïve Bayes, because it is used to estimate the possibility of correct outputs from trust evaluation. As a mainstream supervised learning algorithm, when given a set of training samples, Naïve Bayes is able to analyze the training data and infer the correct output for any valid input data. Thus, when the algorithm of Naïve Bayes is supplied with a node's trust value, we expect it can predict the accuracy of the corresponding output.

(1) **Bayes Theorem**

Bayes theorem is a classic method used to calculate conditional probability; and it provides the statistical foundation for a series of Bayes-based algorithms. It represents the relation between a prior hypothesis H and a prior probability E; this emphasizes the holding possibility of E if there is a higher probability of hypothesis H than $\neg H$ (not-H). Figure 4.8 shows in which group a new triangle (white) should be classified.

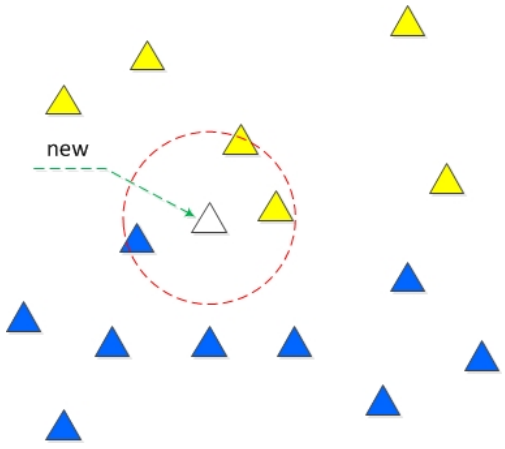


Figure 4.8: A Schematic Diagram of Naïve Bayes Classifier

If X is the unknown data whose class attribute is expected to be classified, H is the beforehand hypothesis used to indicate which class C, the unknown data X, may belong to. For the purpose of classification, we need to determine the probability $p(H \mid X)$,

since it means that the hypothesis H holds, given the observed data X. Then, we are able to express Bayes theorem in a concise way as follows [29]:

$$p(H \mid X) = \frac{p(X \mid H) \times p(H)}{p(X)} \tag{4.4}$$

where *p(H)* is the prior probability of H, and it is unconditional of H. In contrast, the corresponding posterior probability of H conditioned on X is $p(H|X)$, which is the conditional probability because it is derived from the specified H. In the same way, $p(X|H)$ is the posterior probability of X given H (also called *likelihood*). *p(X)* is usually viewed as a normalizing constant. Therefore, Bayes theorem is useful in that it provides an approach to predicting the expected outcome: the posterior probability $p(H|X)$, through *p(H)*, *p(X)*, and $p(X|H)$ [34].

(2) **Naïve Bayes Classifier**

Based on Bayes theorem, the algorithm of Naïve Bayes evolved with the assumption of independence. Apparently, NB is a Bayesian method based on a simple hypothesis in which the presence of a feature is unrelated to other features; and, the influence of features is independent to each other in the hypothesis space [28]. Thus, when sample A owns the features A_i and A_j respectively ($A_i \neq A_j$), the probability of sample A can be classified as B is $p(B|A) = p(B|A_i, A_j) = p(B|A_i) \times p(B|A_j)$.

Abstractly, the probability model for a NB classifier can be described as follows:

(1) In simple terms, the sample B is randomly a Boolean variable; A is a n-dimensional Boolean vector and is represented as $A =< A_1, A_2, \ldots, A_n >$, in which A_i means the ith Boolean randomly attribute. Accordingly to the Bayesian theorem [34],

$$p(B \mid A) = \frac{p(A \mid B) \times p(B)}{\sum_{i=1}^{n} p(A_i \mid B) \times p(B)} \tag{4.5}$$

(2) Assume that there is a target function $f : A \rightarrow B$, the most possible value of B can be deduced through *f(x)*: $\nu_{MAP} = \arg\max_{A_i \in A} p(B \mid A_1, A_2, \ldots, A_n)$

(3) Following the Bayes theorem [7], NB can be represented as *posterior* = *likelihood* × *prior* / *marginal likelihood*. Thus,

$$\nu_{MAP} = \arg\max_{A_i \in A} \frac{p(B) \times p(A_1, A_2, \ldots, A_n) \mid B}{p(A_1, A_2, \ldots, A_n)} \tag{4.6}$$

(4) Since the denominator $p(A_1, A_2, \ldots, A_n)$ represents the value of features A_i, which has been given (this value is often deemed as 1 or a certain constant), it does not depend on B. So

$$\nu_{MAP} = \arg\max_{A_i \in A} p(B) \times p(A_1, A_2, \ldots, A_n) \mid B \tag{4.7}$$

(5) Conditional independence assumes that each feature A_i is independent of other features A_j. Eventually, we can obtain

$$\nu_{NB} = \arg\max_{A_i \in A} p(B) \times \prod_{i=1}^{n} p(A_i \mid B) \tag{4.8}$$

When applying Naïve Bayes classifiers to the real world, NB can be built with real value inputs. It can estimate the outcome for classification in an effectively supervised manner, by virtue of a small amount of training data.

(3) **An Naïve Bayes Example for Sensing Weather Conditions**

With the introduction of Naïve Bayes, we use the following example to elaborate upon how the NB works in a wireless environment. Assume that some sensors are deployed within a specific area, to record simultaneously several meteorological conditions, such as outlook, temperature, and wind. Table 4.2 gives a set of training examples about the weather conditions to determine the recommendations to play tennis outdoors or not.

Table 4.2: Training Examples for Weather Conditions

Day	**Outlook**	**Temperature**	**Wind**	**Play Tennis**
1	Overcast	Mild	Strong	No
2	Sunny	Hot	Weak	Yes
3	Sunny	Hot	Strong	Yes
4	Sunny	Mild	Strong	No
5	Overcast	Hot	Weak	Yes
6	Sunny	Mild	Weak	No
7	Overcast	Mild	Weak	Yes

In this example, *outlook*, *temperature* and *wind* are the features, and *play tennis* is the target classification. Evidently, the problem here is to use the training data combined with these selected features to predict target activity as "*Yes*" or "*No*". When a new object (a new day [Day 8]: *sunny*, *mild*, *strong*) comes, we can make a decision based on the formulated prior probability. First,

Prior probability for $Yes : p(Yes) = 4/7 = 0.5714$

Prior probability for $No : p(No) = 3/7 = 0.4286$

Then we calculate the conditional probability for each feature. For the attribute *outlook*, we have

$p(Sunny \mid Yes) = 2/4 = 0.5$

$p(Sunny \mid No) = 2/3 = 0.6667$

Similarly, for the other two attributes *temperature* and *wind*, they respectively have the conditional probabilities:

$p(Mild \mid Yes) = 1/4 = 0.25$; $p(Mild \mid No) = 3/3 = 1$

$p(Strong \mid Yes) = 1/4 = 0.25$; $p(Strong \mid No) = 2/3 = 0.6667$

Thus, based on the NB classifier, the posterior probability of new weather condition $p(Yes \mid New) = p(Yes) \times p(Sunny \mid Yes) \times p(Mild \mid Yes) \times p(Strong \mid Yes) = 0.5714 \times 0.5 \times 0.25 \times 0.25 = 0.0179$. At the same time, the posterior probability of a new weather condition is $p(No \mid New) = p(No) \mid p(Sunny \mid No) \times p(Mild \mid No) \times p(Strong \mid No)$

$= 0.4286 \times 0.6667 \times 1 \times 0.6667 = 0.1905$. In comparison, $p(No \mid New) > p(Yes \mid New)$, which indicates that it does not recommend playing tennis under such a weather condition, and the new weather condition is classified as *No*.

4.5.3 The Mechanisms of Node Evaluation with Auxiliary Trust

As we have already described, the performance of wireless ad hoc networks depends on the cooperation and trustworthiness of participating wireless nodes. To enhance security in such networks, it is significant to evaluate the trustworthiness of wireless nodes without central authorities. Here, we present our secure node evaluation schemes which manage the procedure of node evaluation in a distributed manner, and make use of cryptographic techniques to offer transmission protection.

The Principles of Node Evaluation

Since the misbehavior of a wireless node is fully unpredictable, the existence of such uncertainty increases the difficulty of misbehavior detection. A well-behaved node may become compromised and so behave maliciously, while an incompetent node may become competent due to environmental changes. For instance, a wireless node may experience some temporary disability at a certain location, due to channel problems, signal interference, traffic congestion or unavailability. As a result, the node in question will not forward packets to other nodes effectively. After moving to a new location where the channel condition is good, the node may recover its state in its new neighborhood. Therefore, node evaluation is significant to the security of the networks, and thus should be dynamic and distributed in wireless ad hoc networks.

Because ad hoc networks lack pre-deployed infrastructure and routing packets are thus transmitted only by relying on intermediate peers, it is worth emphasizing cooperation among wireless nodes. Nevertheless, some nodes may refuse to share their resources and even attempt to benefit from other nodes; and apparently, these nodes can jeopardize the running of a wireless ad hoc network. Based on the characteristics of wireless

nodes, traditional network security solutions cannot be applied well to wireless ad hoc networks, and a wireless node with battery resource constraints cannot act also as a central management center for the entire network to provide robust security management. Therefore, trust is introduced to detect malicious nodes and to prevent misbehavior, and is attributed the following advantages:

- Trust-based evaluation can stimulate wireless nodes to behave well, for instance, by offering incentives to good nodes. Thus, the participants of wireless networks are encouraged to act more responsibly.
- The information associated with trust may help predict the future behavior of a wireless node, which then helps the corresponding node to make decisions accordingly.
- Trust-based information can assist with the detection of selfish or malicious behavior, and furthermore exclude those uncooperative nodes in a network.

Due to the complicated environments of wireless ad hoc networks, it is not sufficient to evaluate a certain node's behavior wholly on the direct observation of the central node. This is because the node probably has objective reasons for which it is unable to contribute to packet forwarding. Nevertheless, under such circumstances, the evaluated node may be honest in other nodes' communities, but the current central node does not know the situation and judges this evaluated node as uncooperative. Hence, reliable and trustworthy neighboring nodes are introduced to the process of node evaluation. First, a central node, which is also the *evaluating node*, will select some trustworthy neighbors as its assistants and then these assistants are required to provide indirect observation of the *evaluated node* upon request from the central node. Subsequently, the central node can employ a supervised learning scheme to measure the reliability of the obtained trust information; then choose different strategies to evaluate the final trustworthiness of the target node with the auxiliary information from its assistants, thus helping the central

node to make proper decisions. The principle of our node evaluation scheme can allow a node to be evaluated more justifiably.

When a node completes the evaluation of other nodes' trust, it does not automatically disseminate this evaluated trust result to the entire network, because such disseminations will incur a heavy overhead and are sometimes redundant. However, the evaluated trust result is available throughout the network to nodes that request it on demand; this enables any node to query other node's trust freely.

Assistant Appointment

When a community is first formed, the central node is able to gather all trust information from its neighbors and then it will choose a trust value as the trust threshold to appoint its trust assistants when evaluating other nodes' trustworthiness. In order to find an appropriate threshold, the central node will find the minimum trust value T_{min} and the maximum trust value T_{max} respectively among its neighbors. Here, let us denote the trust threshold for assistant selection as $TA_{threshold}$ and the number of neighbors as m. Based on the collected trust information, it chooses a trust threshold $TA_{threshold}$ between the T_{min} and T_{max}, so that there will be a certain amount of trustworthy neighbors to be chosen as its assistants.

$$T_{min} < TA_{threshold} < T_{max}$$

Subsequently, all one-hop neighbors whose trust values satisfy the chosen trust threshold are selected as the central nodes' assistants. The following formula describes the process of assistant appointment.

$$T_{N_i} \geq TA_{threshold} \mid N_i \in TA$$

N_i is a neighbor of the central node C. C will check the trust value T_{N_i} of its neighbor N_i to see if N_i satisfies the pre-defined trust threshold $TA_{threshold}$; and, it will then

determine if this neighbor can be appointed as its assistant and stored in its trust assistant list TA accordingly. Here, we formally describe the process of assistant appointment.

Algorithm 4.1 The appointment of a trust assistant by C

$C \rightarrow T_{min}$
$C \rightarrow T_{max}$
repeat
  $C \succ TA_{threshold}$ $\#C$ sets a threshold for its assistants
until $TA_{threshold} \leq T_{min}$ or $TA_{threshold} \geq T_{max}$
for $i = 1 \rightarrow m$ **do**
  if $Trust_{N_i} \geq TA_{threshold}$ **then**
    $N_i \in TA$ $\#N_i$ is included in assistant list
  end if
end for

An Outlier Detection Scheme based on Naïve Bayes

Then, we present an outlier detection scheme which employs Naïve Bayes to determine the accuracy of outcomes from trust evaluation. Beginning with an introduction of outlier detection, we show how our scheme works with secure protection.

(1) **Outlier Definition**

In wireless networks, without the constraints of wires, the number of wireless nodes can increase in a scalable proportion. On the other hand, it is very possible to involve some misbehaving nodes; moreover, regular nodes may behave abnormally in some special scenarios. Obviously, a network will benefit from monitoring data streams and detecting outliers being analyzed. An *outlier* (also referred to as *anomaly*) is usually defined as a distinct observation which differs from other observations [89]. Sometimes, the outlier can be viewed as deviating from observed data and to be inconsistent with other data in a dataset. In the specific scenario, a trust evaluation system undertakes the task of anomaly monitoring and implements a request/reply approach to inquiring the trust information of the evaluated node. The queried node most likely will not provide the

authentic trust value of the evaluated node, so the evaluating node has to use precautious mechanisms to filter incoming falsified trust values. Once a potential outlier is detected, it leads to suspicion that auxiliary information deviates from most of the data recorded, and thereby, the evaluating node is able to provide a prompt reaction to the detected event.

(2) **A NB-based Node Evaluation Scheme**

As we emphasized above, in some cases, direct trust cannot fully reflect the objective trust capacity of a node (*evaluated node*), so the *evaluating node* needs indirect trust from third parties (*queried nodes*) to obtain the recommendations regarding the trust of the *evaluated node*. Thus, indirect trust is able to provide auxiliary information to help evaluate the target node's actual trust. In our scheme, we combine direct trust and indirect trust to evaluate a node's final trust, to avoid the involvement of incorrect trust information when evaluating the node's trust. Figure 4.9 illustrates the scheme to apply Naïve Bayes in the process of trust evaluation.

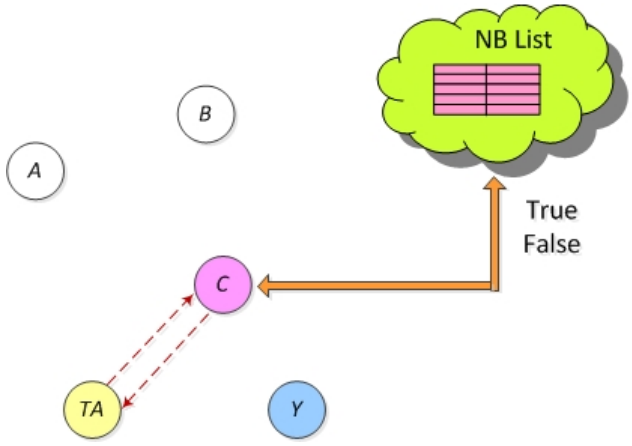


Figure 4.9: The Application of Naïve Bayes in Trust Evaluation

First, a node is assumed to maintain a set of records on its neighbors' past trust PT, and also regarding trust values NT provided by each corresponding neighbor. These historical trust values form a sample space S for our later Naïve Bayes classification. In

the space, each set of trust values (PT and NT) corresponds to a Boolean value of *True* or *False* to indicate whether the provided trust value is true or false, and they are the features selected for NB classification. Thus, we can calculate the prior probability for $True : p(True)$ and $False : p(False)$.

When the evaluating node C wants to evaluate a neighboring node's trust, it will query its trust assistants TA_i about the target node, for example, Y.

$$C \xrightarrow{Query(Y)} TA_i$$

These assistants TA_i will then provide Y's trust values in their individual neighborhoods to the evaluating node C.

$$TA_i \xrightarrow{T(Y)} C$$

After receiving the trust values $T(Y)$ from its assistants TA_i, C will use the Naïve Bayes algorithm to estimate the correctness of the provided trust information based on the aforementioned past trust training data S. During the process of feature selection, two features will be taken into account:

(1) The first feature is the variance of the neighbors' past trust PT. Let us denote M_{PT} as the mean of each neighbor's past trust, and SD_{PT} as the standard deviation of PT. According to the new trust value, C will calculate the variance ν_{TA} of the TA_i's current trust to the mean M_{PT}, to see whether $\nu_{TA} \geq SD_{PT}$ or $\nu_{TA} < SD_{PT}$. For instance, $\nu_{TA} \geq SD_{PT}$; we then calculate the conditional probability for this feature. For the feature $\nu_{TA} \geq SD_{PT}$, we have $p[(\nu_{TA} \geq SD_{PT}) \mid True]$ and $p[(\nu_{TA} \geq SD_{PT})|False)$.

(2) The second feature is the variance of trust values NT provided by each corresponding neighbor. Similarly, we denote M_{NT} as the mean of the trust value provided by each neighbor accordingly, and its standard deviation is SD_{NT}. Thus, C will calculate the variance $\nu_{T(Y)}$ of the received trust value to the mean M_{NT}, to see whether

$\nu_{T(Y)} \geq SD_{NT}$ or $\nu_{T(Y)} < SD_{NT}$. As such, assume that $\nu_{T(Y)} \geq SD_{NT}$, and thereby, we obtain the conditional probability for this feature. In terms of feature $\nu_{T(Y)} \geq SD_{NT}$, we have $p[(\nu_{T(Y)} \geq SD_{NT}) \mid True]$ and $p[(\nu_{T(Y)} \geq SD_{NT}) \mid False)$.

Therefore, based on the NB classifier, C can deduce the posterior probability of the newly received trust value

$$p(True \mid New) = p(True) \times p[(\nu_{TA} \geq SD_{PT}) \mid True] \times p[(\nu_{T(Y)} \geq SD_{NT}) \mid True]$$
$$p(False \mid New) = p(False) \times p[(\nu_{TA} \geq SD_{PT}) \mid False] \times p[(\nu_{T(Y)} \geq SD_{NT}) \mid False]$$

Subsequently, C will compare $p(True \mid New)$ and $p(False \mid New)$. If $p(True \mid New) > p(False \mid New)$, it means that the newly received trust value is reliable; otherwise, if $p(True \mid New) < p(False \mid New)$, the new trust value is not reliable, and it will not be used for the latter un-weighted or weighted NEAT evaluation.

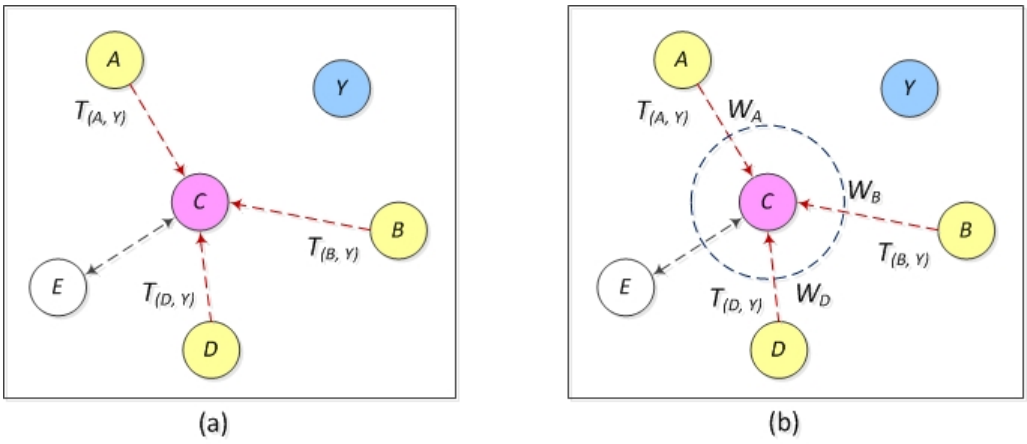


Figure 4.10: Un-weighted Trust Evaluation and Weighted Trust Evaluation

Un-weighted Node Evaluation Scheme

After determining the reliability of trust values from its assistants TA_i, the evaluating node C may choose a variety of methods to compute the evaluated node's final trust: e.g., the average un-weighted means or average weighted means. Figure 4.10 (a) shows the scheme of un-weighted trust evaluation and (b) shows the weighted trust evaluation scheme. First of all, we propose an un-weighted node evaluation scheme (un-weighted NEAT) to assist the central node in evaluating its neighboring nodes' trust. When the central node queries its assistants TA_i about the evaluated node Y, these assistants will then provide the queried node's trust values based on their own knowledge to the central node. If an assistant does not have the trust information of node Y, it will respond with a *VOID* message. After verifying the reliability of the trust values from Naïve Bayes, the central node C uses the average weighted means to measure the node's eventual trust; it then makes the corresponding decision of whether to keep the node in the community or to exclude it. The following formula evaluates the node's eventual trust:

$$T_Y = \frac{T_{(C,Y)} + \sum_{i=1}^{n} [T_{(TA_i,Y)} \mid TA_i \in TA]}{n+1} \tag{4.9}$$

where $T_{(C,Y)}$ is the trust value of the central node C for the evaluated node Y, $T_{(TA_i,Y)}$ is the trust value of the assistant TA_i for the same node Y, and n is the number of trust assistants that reply with a valid message. Thus, these assistants can help prevent the scenario where a central node is a potential attacker or a malicious node, because the central node does not need to make its decision here based only on its own information. Using the information of the assistants, the central node can make more reliable and objective decisions when evaluating a node's trust. The following algorithm shows the procedure of un-weighted node evaluation.

Weighted Node Evaluation Scheme

Then, we propose a weighted node evaluation scheme (weighted NEAT) that also utilizes collected trust information from the assistants of the central node to help it better evaluate its neighboring nodes' trusts.

Algorithm 4.2 Un-weighted node evaluation

$C \succ Msg(Y)$ #operation at C
for $i = 1 \rightarrow k$ **do**
  $C \xrightarrow{Msg(Y)} TA_i$
end for #C composes a query to assistants
#operation at TA_i
for $i = 1 \rightarrow k$ **do**
  if $Y \subset TA_i$ **then**
    $TA_i \xrightarrow{Trust(Y)} C$ #TA_i replys *Trust(Y)*
  else
    $TA_i \xrightarrow{VOID} C$ #TA_i replys *VOID*
  end if
end for
$T_Y = Trust(C)$ #operation at C
$n = 1$
for $i = 1 \rightarrow k$ **do**
  if $Trust(Y)_{TA_i} \neq VOID$ **then**
    $T_Y = T_Y + Trust(Y)_{TA_i}$
    $n++$
  end if
end for

Let us assume that for the central node C, when C wants to evaluate a neighboring node's trust, it will query its appointed assistants TA_i about this neighboring node Y. If the number of assistants able to reply to the trust query, with effective responses, is still n, then these n assistants will provide C with the trust values of node Y in their individual communities. Afterwards, C uses the weighted means to measure the node's eventual trust, making a decision accordingly. The following formula first calculates the average trust value T_{AVG} from the set S_{TA} of assistants.

$$T_{AVG} = \frac{T_{(C,Y)} + \sum_{i=1}^{n} [T_{(TA_i,Y)} \mid TA_i \in TA]}{n+1} \tag{4.10}$$

where $T_{(TA_i,Y)}$ is the trust value of the assistant TA_i to the specified node Y. Formula 4.11 then computes the weight w_i of each assistant based on its own trust value in the entire set of trust assistants TA:

$$w_i = \frac{T_{(TA_i,Y)} \mid TA_i \in TA}{T_{AVG}} \tag{4.11}$$

Formula 4.12 evaluates the node's eventual trust T_Y.

$$T_Y = \frac{T_{(C,Y)} \times w_C + \sum_{i=1}^{n} T_{(TA_i,Y)} \times w_i}{n+1} \tag{4.12}$$

where $T_{(C,Y)}$ is the trust value of the central node C for the same node Y, and w_C is 1 for the central node because w_C is taken as a standard to measure the importance of trust evaluations from other assistants. The algorithm below describes the process of weighted node evaluation. Thus, the central node does not need to make its decision based only on its own knowledge. Even if a node receives a malicious evaluation, the central node will employ NEAT mechanisms to evaluate the nodes trust, and the effects of the malicious evaluation would be mitigated because the other assistants would offer objective and accurate evaluations.

Detection of Malicious Neighbor

In a community, the central node can rely on trust evaluation to detect malicious nodes. The central node will first set a trust threshold for detecting malicious neighbor T_{not}, this will indicate the minimum trust value that a node can have in order to take part in communication as a secure member or to survive within its community. After the central node obtains the final trust evaluation of the target node based on our NEAT

mechanisms, it will judge if the final trust value is below the trust threshold for detecting malicious neighbors and then determine if the target node can still be a member of its community. The following formula shows the procedure of detection.

Algorithm 4.3 Weighted node evaluation

$C \succ Msg(Y)$ #operation at C
for $i = 1 \rightarrow k$ **do**
  $C \xrightarrow{Msg(Y)} TA_i$
end for #C composes a query to assistants
#operation at TA_i
for $i = 1 \rightarrow k$ **do**
  if $Y \subset TA_i$ **then**
    $TA_i \xrightarrow{Trust(Y)} C$ #TA_i replys *Trust(Y)*
  else
    $TA_i \xrightarrow{VOID} C$ #TA_i replys *VOID*
  end if
end for
$T_{AVG} = Trust(C)$ #operation at C
$n = 1$
for $i = 1 \rightarrow k$ **do**
  if $Trust(Y)_{TA_i} \neq VOID$ **then**
    $T_{AVG} = T_{AVG} + Trust(Y)_{TA_i}$
    $n++$ #calculate the average trust from TA_i
  end if
end for
$T_{AVG} = T_{AVG}/n$
for $i = 1 \rightarrow k$ **do**
  $w_i = Trust(Y)_{TA_i}/T_{AVG}$
end for #calculate the weight of each assistant
$w_C = 1$
$T_Y = Trust(C) \times w_C$
for $i = 1 \rightarrow k$ **do**
  $T_Y = T_Y + Trust(Y)_{TA_i} \times w_i$
end for
$T_Y = T_Y/n$

Thus, if T_Y is greater than or equal to the pre-defined threshold T_{not}, then the target node Y can communicate with C or stay in community C; otherwise, Y will be deemed as an untrustworthy node and excluded from community C.

$$\begin{cases} T_Y \geq T_{not}, & Y \in C \\ T_Y < T_{not}, & Y \notin C \end{cases}$$

As for the selection of threshold T_{not}, some current work [62, 63] indicates that it is not easy to determine T_{not}. A node may be judged as honest in a community; whereas, it may be deemed as unqualified in another community if these communities do not have the same threshold. Actually, in our mechanisms, we employ such a mechanism to encourage different communities to set different thresholds, because each community has various situations and each node may behave differently in different communities. A central node is able to collect all the trust information of its neighbors; thus, the central node can calculate an average value among its neighbors as its threshold T_{not}, so that $T_{min} < T_{not} < T_{max}$. On the one hand, those neighbors which own lower trust values cannot be included in the community; on the other hand, a certain amount of neighbors with a higher trust value above T_{not} are included in the community, so that they can work for the packet forwarding of the central node. Obviously, those nodes with higher trust values than T_{not} are more trustworthy than those unqualified neighbors.

4.6 Security and Performance Analysis

4.6.1 Security Semantics and Analysis

Security Analysis of TOMS

In this section, we provide a formal analysis of our trust system and verify that the goals described above are achieved in our system. The methodology related to security analysis [16, 40, 82] is utilized here. The basic notations are as follows [21]: n represents a principal, where N^* represents a set of participating principals and an individual node $n \in N^*$. The current beliefs of the participating principals are deduced by their initial states and possessions. In particular, S is a principal and is also the sender, and R is

another principal and is designated as the intended receiver. A malicious node is denoted as M_N, and a cooperative node is denoted as N_{CO}. PK_S and PK_R are respectively the public keys of the sender and receiver.

Lemma 1. *If $n \in C^*$, and $n \succ F(c)$, then n is a well-behaved node.*

Proof. For a node n that is a member of the community C^*, if the node n implements the function $F(c)$ designated by the community C^* and confirms the specifications of the system, then the node n is reckoned as a well-behaved neighbor of the central node c.

Lemma 2. *If $n \in C^*$, and $n \prec F(c)$, then n is a bad-behaving node.*

Proof. Similarly, for the member n of the community C^*, if it does not follow the specifications of the system and finish the function $F(c)$ designated by the community C^*, then the node n is judged as a misbehaving neighbor of the central node c.

Theorem 1. *In TOMS, any central node is able to classify its neighbors based on their past behavior and reputation.*

Proof. In one community $C^*(c, n)$, the central node c always monitors its neighbors by means of acknowledgements, or overhears any other node n within its transmission range.

(1) According to Lemma 1, for a well-behaved neighbor n, it can help to relay the traffic for the central node c, so it will be rewarded with the increase of its trust value T_n. Thus, the more it helps to forward messages for the central node c, the higher its trust value T_n becomes, so that it is easier to be clustered into the class of well-behaved nodes.

(2) Similarly, based on Lemma 2, for the misbehaving neighbor n, if it does not obey the rules of the system in the community C^*, it will be punished with the decrease of its trust value T_n. Thereby, this node is more likely to be clustered into the misbehaving cluster.

(3) For those neighbors whose trust values are between the misbehaving and well-behaved thresholds, they are considered as a cluster in which the nodes are not malicious

nodes, but they have not contributed much to relaying traffic. Consequently, they belong to the cluster of moderate nodes.

Theorem 2. *In TOMS, trust can be updated sensitively for nodes based on their past behavior and reputation.*

Proof. The trust computation in TOMS is unique and effective because it employs different mathematical functions to simulate the change in trust based on different categories of nodes.

(1) For those nodes that are classified into the well-behaved cluster, TOMS uses an exponential function to calculate the changes in trust. In particular, the exponential function involves the historical evaluation of an individual node into the exponent, so the trust model will have a fast increase in the trust of well-behaved nodes. This matches the theory of giving greater rewards to well-behaved nodes.

(2) For the nodes that are classified into the malicious cluster, TOMS uses a logarithmic function to describe the changes in trust. In particular, the logarithmic function also involves the historical evaluation of an individual node into the logarithm, so that the trust model will provide a fast decrease in trust to malicious nodes. In comparison to well-behaved nodes, this matches the theory of punishing more quickly for malicious nodes.

(3) TOMS utilizes a linear-shape function to simulate the change in trust for those nodes with medium trusts. It is reasonable to make a linear-like increase for these nodes.

Lemma 3. *If* $n \in C^*$, *and* $T_n > T_{good}$, *then* $C^* \triangleright n$.

Proof. Here, node n is a member of the community C^*. The trust value T_n of the node n exceeds the trust requirement T_{good} of the well-behaved cluster, so it is thought to be a good neighbor and is kept in the community C^* with the evaluation of an exponential function.

Lemma 4. *If* $n \in C^*$, *and* $T_n < T_{bad}$, *then* $C^* \triangleleft n$.

Proof. Similarly, for the member n of the community C^*, the trust value T_n falls below the trust threshold T_{bad} of the misbehaving cluster, then this node n does not have

a good reputation in community C^* and will apparently be evaluated in the community C^* with a logarithmic function.

Theorem 3. *TOMS can identify malicious behavior and exclude malicious node.*

Proof. In TOMS, trust is an effective approach for detecting malicious nodes and our trust model is efficient in calculating the trust of each individual node.

First, based on the notion of trust, each node is associated with a reputation criterion to evaluate its behavior. The critical part of our system is that the evaluation is not performed by the individual node itself, but by the other node that it serves. Thus, it avoids the possibility that the individual node overestimates its contribution to other nodes, so that the evaluation is objective.

Second, the central node in a community will always monitor their behavior and overhear its neighbors. Then it will calculate the trust of each neighbor respectively based on their corresponding actions. If any malicious behavior occurs, it can be detected by the central node and reflected in the TOMS computation model.

Third, TOMS will set respective thresholds for malicious nodes and honest nodes. Once the trust value of a node falls into the threshold for malicious nodes, it will be judged as a malicious node and thus its trust will be computed based on a logarithmic function that is used only for malicious nodes; however, if the trust value of the node exceeds the threshold for well-behaved nodes, it will be treated as an honest node and thereby its trust will be calculated based on an exponential function that is designed especially for honest nodes.

Fourth, once a node is judged to be a malicious node since its trust value is too low, it will be excluded from the central nodes community and recorded in the corresponding black list, so that it will not be able to join the same community later. Therefore, a node in TOMS can identify a neighbors malicious behavior and malicious nodes will be excluded from the community.

Theorem 4. *NEAT guarantees that each evaluation related to trust is objective and mitigates the justifiable evaluations from malicious nodes.*

Proof. It is possible that some nodes mistakenly evaluate the trust of other nodes. For example, some nodes may overestimate the contributions of others, including malicious nodes if the collusion of nodes happens, and some nodes may underestimate the contributions from other nodes. In order to avoid malicious nodes deliberately giving other nodes overly-high or overly-low trust evaluations, the proposed NEAT can effectively prevent it and provide non-prejudiced evaluation, because NEAT not only requires the evaluation from the central node for a certain node, but also requires the evaluations from as many other nodes as possible for the same node. Furthermore, the final trust value of the node is calculated based on all trust values that are obtained from all relevant assistants. Thus, the trust value from one node can no longer determine the reputation evaluation individually.

Security Analysis of NEAT

Since malicious nodes are clearly harmful to a presently healthy network, it is essential to evaluate a node's behavior and trustworthiness. Sometimes a well-behaved node may become compromised due to a variety of unforeseen reasons and *vice versa*. As a consequence, node evaluation is dynamic and distributed in wireless networks, and it is important for the security of networks as well [100]. A realistic trust evaluation is able to help the central node determine which nodes should be kept and which nodes should not be included in its community. In our current work, we concentrate on the prevention of trust-oriented attacks, because unrealistic or deviated trust information can disturb the procedure of trust evaluation and affect the rational result of trust evaluation. Here, the notations and security statement follows the specifications as we described in Sections 3.4.1 and 4.6.1.

Internal attacks come from those compromised nodes inside a network, which are difficult to detect and prevent. Once the compromised node is chosen as a central node's assistant, it intends to disturb the procedure of the central node's trust evaluation by providing a deviated auxiliary trust evaluation to mislead the central node. In the

standard attack model used in secure trust evaluation in wireless networks [4, 124], an attacker launches the following attacks, all of which attempt to compromise the content of trust evaluation in terms of trust and node evaluation.

(1) *Increasing Evaluation Attack*: after a malicious node M_N is selected as the trust assistant of a central node, it may intentionally exaggerate the contributions of a certain node Y and thereby this node can obtain higher trust. Such an attack obviously overestimates a node's trust.

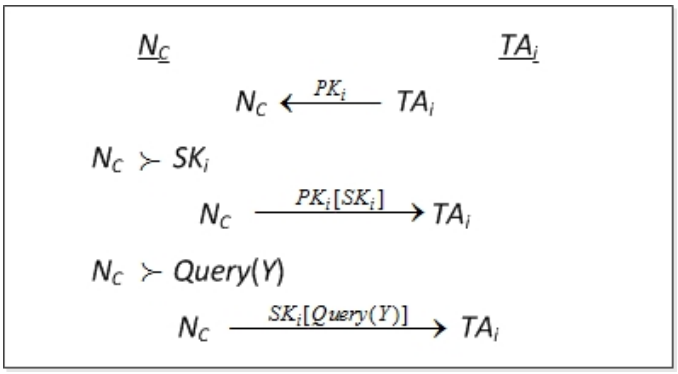
(2) *Decreasing Evaluation Attack*: likewise, the malicious node can also issue incorrect trust information that underestimates other honest nodes' contributions; this gives prominence to a malicious node's own trustworthiness in a network, making it seem more honest and reliable.

(3) *Random Evaluation Attack*: the malicious nodes M_N can randomly generate a random value instead of the correct trust value, for the above reasons.

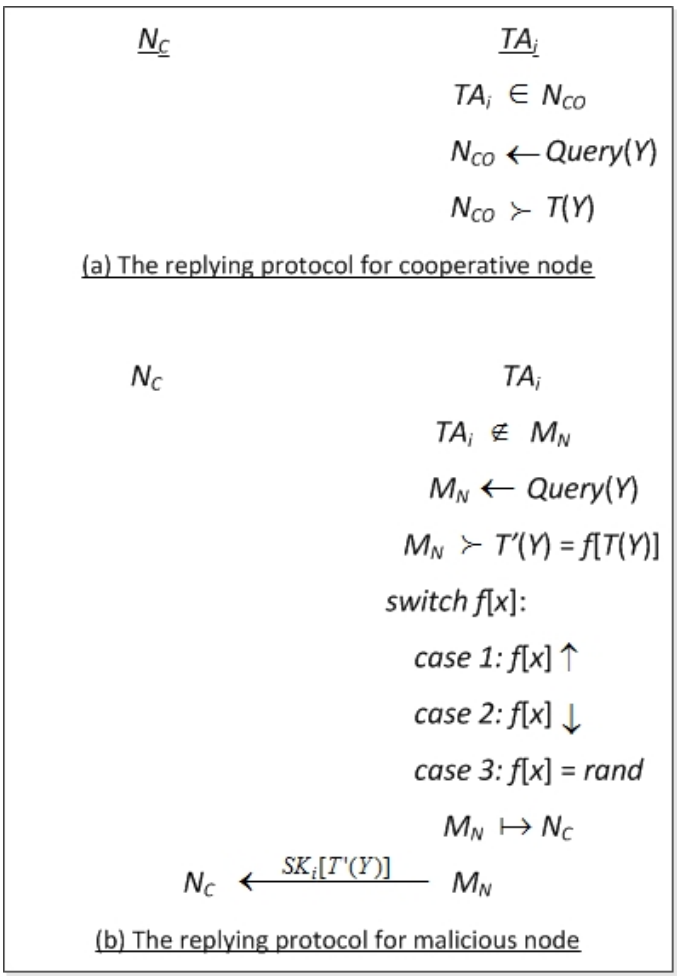
Issuing and Replying Protocols

Either by increasing evaluation attack or in decreasing evaluation attack, the attacker responds as a compromised assistant TA_i' of N_C and still provides secondary trust information to N_C. We respectively analyze how our NB detection scheme is secured along with the issuing and replying protocols.

At the beginning of issuing a trust query to its assistants, N_C and its trust assistant TA_i are supposed to have already exchanged their public keys PK_{N_C} and PK_{TA_i}. Afterwards, N_C will first generate a secret key SK_{TA_i} for its neighbor TA_i, and distribute this key to TA_i using the public key PK_{TA_i} of its corresponding neighbor. Subsequently, N_C will send a query message Query(Y) to TA_i which includes the queried node's identifier ID_Y, which is encrypted by the SK_{TA_i}. After TA_i receives such a message, it can decrypt this message using the corresponding key SK_{TA_i}. Thus, the issuing process of the trust query message is secured through symmetric key cryptography.



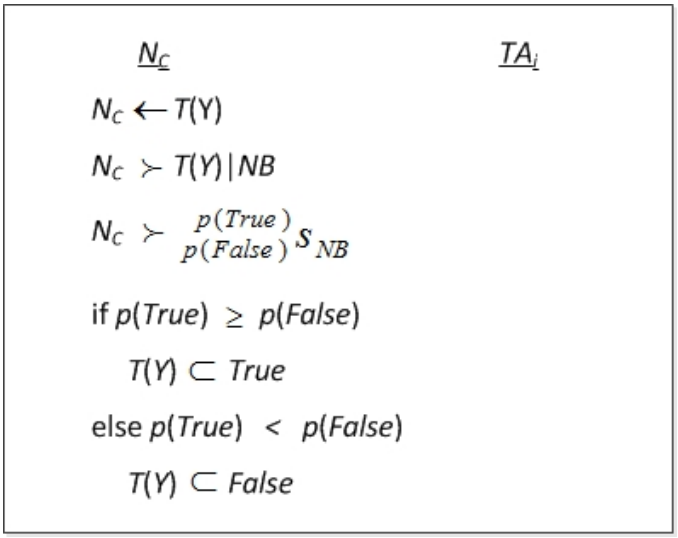
The Issuing Protocol



The Replying Protocol

When the assistant TA_i of N_C receives the trust query message Query(Y), TA_i will respond this message to compose a trust reply message which includes the requested trust $T(Y)$ and then send it back to N_C. If TA_i is a cooperative node N_{CO}, it will reply N_C with the true $T(Y)$ and the message is also protected by SK_{TA_i}. Otherwise, if TA_i is a malicious node M_N, it may make use of three different approaches to falsify the requested trust $T(Y)$ as $T'(Y)$: (1) increase $T(Y)$; (2) decrease $T(Y)$; (3) choose a random value instead of $T(Y)$. Then, M_N will follow the same procedure to encrypt $T'(Y)$ with SK_{TA_i} and reply it to N_C.

Once N_C obtains the feedback from TA_i, it will employ Naïve Bayes classifier to infer the authenticity of the provided trust value $T(Y)$. Through evaluating the TA_i's trust and this provided $T(Y)$ based on the training data over the NB hypothesis space S_{NB}, it acquires the probabilities $p(True)$ and $p(False)$. Furthermore, N_C can determine whether $T(Y)$ is trustworthy or not and thereby take further actions.



The Verifying Protocol for Malicious Evaluation

Thus, after the filter of NB classifier, less unrealistic trust information from malicious nodes can probably be released to the trust evaluation process. However, based on our NEAT schemes, even if some unrealistic or falsified information is involved in trust eval-

uation, these attacks can be alleviated by our proposed NEAT schemes. We continue to analyze how our schemes offer the security relief and use weighted NEAT as an example.

$$N_C \xleftarrow{Query(Y)} TA_i \in TA$$
$$N_C \succ NEAT:$$
$$N_C \xrightarrow{Calculate} w_i \ and \ w_{M_N}$$
$$N_C \xrightarrow{Evaluate} Average(w_i \times TA_i^Y) \qquad (including \ w_{M_N} \ and \ T_{M_N}^Y)$$

Here, though the central node N_C receives the overestimated trust evaluation from the attacker M_N, it also gets true evaluations from other assistants TA_i. Then N_C applies the weighted NEAT mechanism and calculates the eventual trust value of node Y based on all information that it collected. Consequently, N_C can obtain a closer value to the actual trust value of node Y, compared to the exaggerated value provided by the malicious assistant M_N. Thus, this mechanism greatly mitigates the malicious influence of exaggerated trust evaluation from attackers. As such, the underestimated trust evaluation does not also greatly influence the final trust evaluation of node Y with the use of our weighted NEAT mechanism. In comparison to the underestimated value replied by M_N, the central node N_C is able to receive a closer value to the actual value of node Y under such circumstance as well.

Since symmetric encryption algorithm is applied in the communication between the central node N_C and its neighbors N_i, the attacker M_N cannot decipher the intercepted trust reply message even if it is able to eavesdrop on the communicated messages. Both the central node N_C and its assistant TA_i will use the secret key SK_{TA_i} to encrypt their trust query and reply messages, and the secret key SK_{TA_i} is only known between them. M_N does not have any knowledge about SK_{TA_i} because this key is not shared in the community C at all. Therefore, in our community model, the one-to-one secret key mechanism can effectively prevent the attack of modification.

As for the accuracy of evaluation, the central node N_C is able to obtain a more

accurate final trust of the targeted node Y. Suppose that Y has a real trust value TT_Y for its own behavior, and the direct observation of N_C about Y has deviation δ_{N_C} from the true trust TT_Y. The auxiliary information from N_C's assistants can then help correct this deviation δ_{N_C} and the accuracy of evaluation is as follows:

$$
\begin{aligned}
&|T_{(N_C,Y)} - TT_Y| = \delta_{N_C} \\
&N_C \xrightarrow{Query(Y)} TA_i \in TA \\
&N_C \xleftarrow{T(Y)} TA_i \\
&|TA_{(TA_i,Y)} - TT_Y| = \delta_{TA_i} \\
&\delta_{TA_i} < \delta_{N_C} \\
&N_C \succ NEAT : \delta_Y = \delta_{N_C} + \sum_{i=1}^{n} \delta_{TA_i}, \delta_Y < \delta_{N_C}
\end{aligned}
$$

After N_C queries its assistants TA_i and gets their responses $T_{(TA_i,Y)}$, N_C can sum up the auxiliary trust values together with its own trust estimation. Since the assistant TA_i has a higher trustworthiness, the information that TA_i provided is more reliable than the random query among the neighbors N_i of N_C. Therefore, the deviation δ_{TA_i}, obtained between the assistant's estimation to Y and the true trust TT_Y, is smaller. When N_C applies NEAT mechanisms, it involves the smaller deviation into the final evaluation of Y, which is smaller than the original deviation δ_{N_C} and thus is closer to the true trust TT_Y. Since the trust of the central node to node Y is considered to evaluate the final trust of Y, the deviation δ_Y is smaller than the deviation obtained just from the assistant $\delta_{TA_M}(T_{TA_M} - TT_Y)$. In conclusion, the final evaluation is more accurate, compared to the situation where only the central node N_C attempts to calculate Y's trust value with its own information, and therefore, the outcome of trust-based node evaluation is more trustworthy and reliable.

4.6.2 Performance Evaluation

We carried out an extensive set of experiments within a wireless and mobile environment based on Network Simulator *ns2* [49]. The experimental environment is set as we described in Section 3.4.2. We have chosen the following performance metrics for evaluating our trust system:

(1) *The Size of Network*: the total amount of the nodes in a network, including each central node and all of its neighbors;

(2) *Connectivity*: the percentage of the sent requests successfully answered by the destination, out of the requests sent by the source;

(3) *Percentage of Malicious Nodes*: the percentage of malicious nodes refers to the percent of malicious nodes that take part in communication within the entire community, while the ratio of malicious nodes indicates the proportion of malicious nodes that are included in the communicating process;

(4) *Security Overhead*: the ratio of the number of messages in relation to security and trust, to the total number of packets used for the formal communication among all nodes.

Linear Trust Evaluation Models

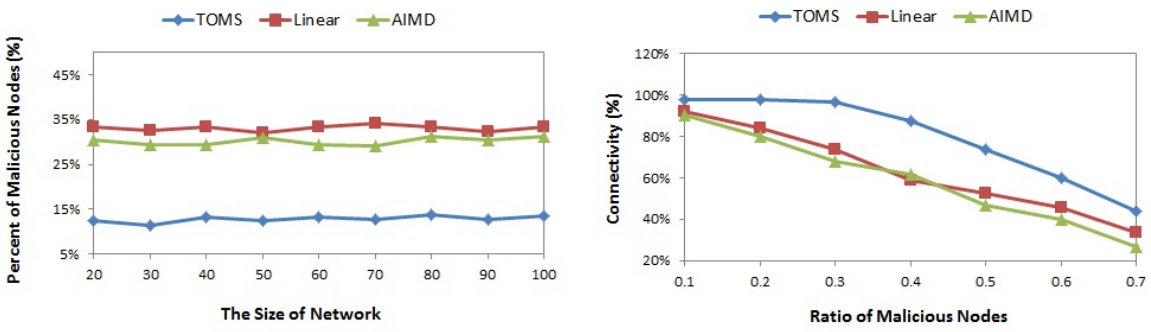
We compare TOMS with the already accepted trust schemes in [27, 81, 105], all of which use mainly the linear function or probabilistic approaches to measure the changes in trust in a wireless context. Thus, we use the following linear function to describe the trust between the interactions of nodes: $T = \alpha \times x + T_0$, where x represents an action of one node and T_0 is the initial trust value. We named this linear trust evaluation model as *Linear*. In addition, we also abstract another additive increase and multiplicative decrease trust model, in which the nodes' trust changes are evaluated based on an additive

increase for a successful report and a multiplicative decrease for a failed report. The following functions are used to describe the trust within the interaction of nodes.

Cooperative behavior: $Trust_{Current} = Trust_{Recent} + \alpha$

Uncooperative behavior: $Trust_{Current} = \beta \times Trust_{Recent}$

where α and β are the respective scaling factors for successful behavior and unsuccessful behavior. We call this model as *AIMD*. Based on these traditional trust schemes, the nodes are managed based on the trust values and thereby form different groups: the values for differentiating these groups are empirically selected as 0.3 and 0.6. As a result, these systems can realistically reflect the basic trust schemes according to most of the existing trust systems.



(a) (b)

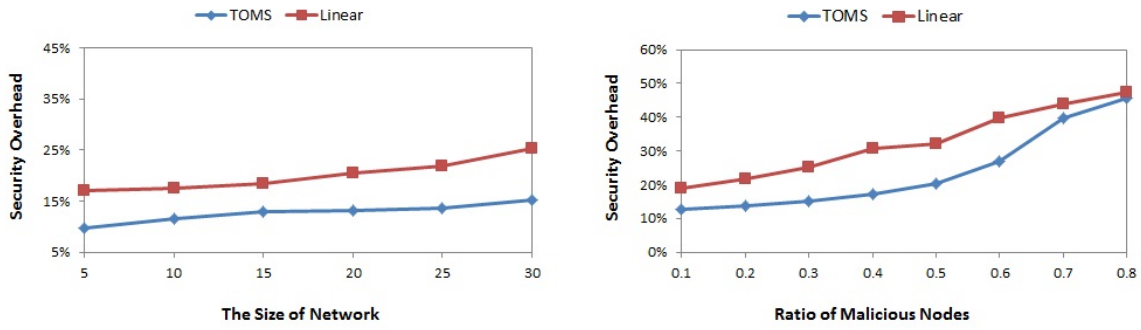
Figure 4.11: Percentage of Malicious Nodes and Connectivity

Performance Analysis of TOMS

Figure 4.11 present comparisons of the factors affecting the percentages of malicious nodes involved in the TOMS and linear systems. These graphs show that trust requirement does affect the percentage of malicious nodes, and the percentage of malicious nodes also affects the connectivity in different systems. In TOMS, the percentage of malicious nodes indicates that fewer malicious nodes are included in the course of forwarding messages for the central node; this shows that taking the exact trust value as

the trust requirement plays an important role in this regard. At the same time, it is reasonable that TOMS performs better than the linear trust system even though the variations for the malicious nodes reveal the same trends.

Figure 4.12 (a) shows that TOMS has a lower security overhead used for community management than that of the linear trust system based on the size of network. This is mainly because our TOMS system does not classify all neighbors into different groups, and it only needs to communicate with individual neighbors each time. Unlike the group-based linear system, each node clusters all of its neighbors into three groups. This means that, if a neighbor changes its reputation value from one level to another, each group needs to be updated accordingly. Figure 4.12 (b) compares the security overhead spent on the TOMS and linear systems with the increase of the percentage of malicious nodes, showing that TOMS has the same increase trend on security overhead when compared to the linear system. It is understood that TOMS will incur a somewhat lower cost since it adopts a more precise manner of managing the community, whereas the linear system classifies the nodes more roughly.



(a) (b)

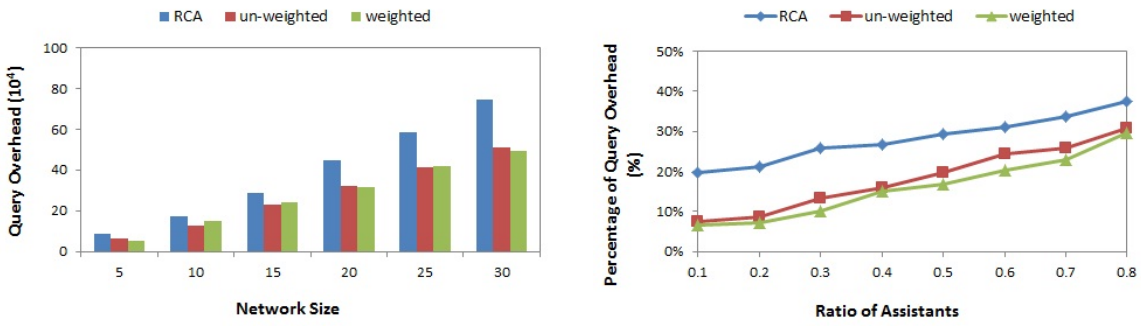
Figure 4.12: Security Overhead based on Network Size

Performance Analysis of NEAT

In addition, we also evaluate the performance of our proposed NEAT schemes, in comparison to a baseline strategy with help from randomly chosen neighboring assistants.

Our proposed schemes can derive a neighboring node's trust with higher accuracy so that the influence of malicious behavior on trust evaluation is greatly mitigated. On the other hand, a comparable node evaluation scheme that we refer to as RCA, is abstracted from some already accepted node evaluation systems which take advantage of random methods of choosing some nodes in the neighborhood of the target node to provide secondary trust information in a network.

Overall, the experiments are divided into two phases, in order to evaluate the performance of our schemes and observe their behavior in wireless ad hoc environments. First, we measure the overhead spent on node evaluation in our schemes. Next, other experiments examine the effectiveness of mitigating malicious nodes' influence on node evaluation, as well as the accuracy of node evaluation of our schemes. We have chosen the following performance metrics to measure our node evaluation schemes: 1) *Query Overhead*: records the number of packets used to send trust-related query messages by the central nodes. 2) *Percentage of Query Overhead*: indicates the percentage of traffic used to send packets for node evaluation, out of the total amount of traffic. 3) *Query Accuracy*: derives from the trust values sent by the queried assistants in a community and reflects the accuracy of other nodes recommendations.



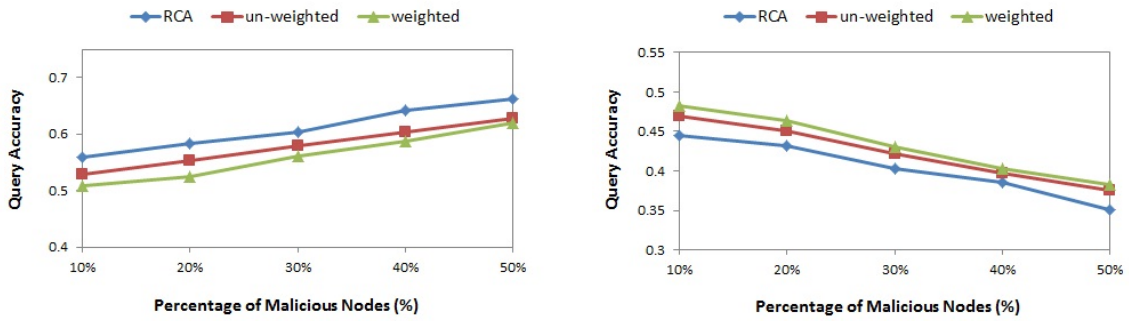
(a) (b)

Figure 4.13: Query Overhead *vs.* Number of Assistants

Firstly, we investigate how communication costs will change in all node evaluation schemes, including RCA and NEAT. In the two figures below, both reflect the influence

of network density on our mechanisms. Figure 4.13 (a) shows that the number of packets used for evaluating trust increases with network size in all schemes. This is because the more assistants that are selected, the more packets are sent between the central node and its assistants for trust query. Additionally, the network size increases, so the amount of communities becomes larger as well. Moreover, the query overhead for RCA is higher than the overhead of un-weighted NEAT and weighted NEAT. Figure 4.13 (b) also shows that the percentages of trust evaluation packets in both NEAT schemes are obviously lower than in RCA when the number of assistants used for node evaluation increases.

In these two graphs, for whichever NEAT scheme is used to evaluate un-weighted or weighted nodes, NEAT can result in a lower cost than RCA in all scenarios. Our NEAT schemes thus outperform RCA in terms of trust evaluation overhead. This can be explained by the NEAT schemes only choosing the neighboring nodes with higher trust values as trustworthy assistants. The RCA scheme, however, employs a fully random strategy to choose assistants. Uncertainty exists in the selection of assistant nodes for RCA and evaluation overhead thereby increases. At the same time, we also note that un-weighted and weighted NEAT schemes have almost the same performance on the query overhead. This is understandable because both of them use the same mechanisms to appoint assistants, so their performance is similar.



(a) Query Accuracy based on Exaggeration Attack (b) Query Accuracy based on Underestimation Attack

Figure 4.14: Query Accuracy *vs.* Number of Assistants

Figure 4.14 (a) illustrates that the value of trust query from NEAT schemes is lower

than that from RCA with the increase of malicious nodes, where the evaluated node actually has a medium trust value and all malicious nodes issue an exaggerated trust value to respond to the corresponding trust query. Figure 4.14 (b) also demonstrates that the value of trust query from NEAT schemes is higher than that from RCA as the number of malicious nodes increases, where all malicious nodes issue an underestimated trust value to reply the relevant trust query and the evaluated node actually has a medium trust value. This means that NEAT can obtain a more accurate trust value than the randomly chosen assistant scheme, since more assistants which are also more trustworthy join the trust query process and give an objective trust evaluation. Thus, NEAT can obviously mitigate the influence of malicious evaluation from malicious nodes and acquire a more realistic trust value. From these two figures, we can see that the tendency for accuracy matches our design principle: the more assistants there are, the more accurate the evaluation is, based on a fixed proportion of malicious nodes. These malicious nodes will initiate increasing evaluation attacks or decreasing evaluation attack. Also, we find that weighted NEAT has a slightly lower trust evaluation than un-weighted NEAT in Figure 4.14 (a), and has a slightly higher trust evaluation than un-weighted NEAT in Figure 4.14 (b). This is because the weighted NEAT uses weighted means to measure the node's final trust and the importance of trust value provided by each assistant is measured according to the reliability of the corresponding trust value. Still, we have to acknowledge that the difference between them is not obvious.

4.6.3 Discussion

The proposed TOMS system is unique and has a number of characteristics. These features form a flexible, intelligent and adaptive trust-based security system. We will enumerate the main characteristics and analyze their advantages.

- *Suitablity for Wireless Devices*: TOMS is especially designed for wireless-oriented devices. We can note that our trust model is completely different from other trust

schemes: it only employs flexible but simple mathematical models and mechanisms to simulate the changes of each node's trust. Since many wireless or mobile devices are portable or embedded, they all have low-processing ability and require instant information exchange, so the complicated computational models will consume the valuable resource and delay the real-time communications. For example, within WSNs the sensors generally need energy-efficient algorithms in order to avoid the exhaustion of their batteries, and so our trust model simplifies the computation of trust but still makes the evaluation of trust promptly and adaptively. Thus, our trust computation model is suitable for wireless and mobile computing.

- *Adaptive Evaluation*: An update mechanism similar to AODV [86] is combined with our trust evaluation system, which provides the flexible and adaptive management of trust update. When a central node evaluates its neighbors, the evaluated trust value is not permanently kept as the neighbor's trust record. As the periodically broadcast *Hello* messages, which indicate an ending of the preceding session and the beginning of the following session, the current trust value of a specific neighbor will be used as the recent trust value of the same neighbor. Repeatedly, the central node will always obtain the latest trust value for its neighboring node's reputation.

- *Intelligent Evaluation*: In TOMS, each node is associated with its previous records related to its behavior, and the records are kept by each central node as trust to avoid any tampering with the reputation evaluation. In comparison to using the linear function to simulate the changes in trust, our system employs different functions to simulate different types of nodes. For example, the exponential shape corresponds to those nodes associated with fast increase in trust, the almost linear shape is used for those nodes associated with moderate trust change, and the logarithmic shape corresponds to those nodes associated with slow increase in trust. Therefore, it avoids the onefold calculation for various categories of nodes and involves intelligent computing to evaluate the trust of nodes.

- *Trust-based Classification*: In a community, the central node does not strictly classify its neighbors according to their trust levels, such as *High*, *Medium* or *Low*. When different neighors have different trust values, their trust computation functions are different as well. In addition, their trust classification will be updated in each session, so the trust-based classification is completely dynamic and flexible.

Based on the aforementioned NEAT mechanisms (un-weighted or weighted), we can learn that the more assistants a central node has, the more accurate the eventual trust evaluation is. Apparently, our proposed mechanisms are suitable to such an environment where a majority of nodes are honest while a minority of nodes is compromised, and there is no collusion between those compromised nodes. Based on our NB detection scheme, quite a few attacks can be alleviated due to the combination of direct and indirect trust evaluation, and even some attacks can be avoided by the effective use of cryptography.

On the other hand, it can be noted that our model does not dramatically decrease a node's trust but lessen its trust gradually. This is due to the fact that we consider that some uncooperative behavior may be caused by objective reasons, for example, channel congestion, interference, etc. Both TOMS and NEAT models allow the existence of a certain light-weighted uncooperative behavior. Therefore, by introducing our trust evaluation model, which is not complicated but adaptive, these models can be applicable for wireless environments.

4.7 Summary

The issue of trust has obtained wide attention and become a challenge in wireless and ad hoc networks, due to its complexity. In this chapter, we present the concept of trust and formulate the theory of trust and community. Furthermore, we avoid the complexity of the traditional trust model, and propose a novel trust model that can effectively compute the trust value of a node. Also, our trust management scheme is able to manage nodes efficiently. In TOMS, the nodes throughout the entire network are

managed in a distributed, dynamic, and efficient manner. In addition, we also explore the issue of outlier detection and recognize the capability of trust as an effective tool to monitor the node's activities in a network. Thereby, we study the feasibility of applying the Naïve Bayes classifier to solve the issue of predicting the reliability of indirect trust information. Details also focus on the modeling and analysis of the node's misbehavior in the network.

Through the security analysis of our system, TOMS offers provable security properties. Moreover, we present an analytical framework to prevent misbehavior aimed to the process of trust evaluation. The analysis and discussion demonstrate that Naïve Bayes can generate reliable predictions. Both of these trust-based security mechanisms demonstrates that the proposed trust computation and management solution offers a new and effective computing paradigm for guaranteeing secure and reliable communications.

Chapter 5

HiC: A Hybrid Cryptosystem for Data Protection

In a wireless network, open and wireless communication channels replace wired connections, because wireless signal acts as the platform of data transmission. However, the open and shared nature of wireless networks also causes a number of security concerns. In particular, data confidentiality is one of the most important issues. As usual, the technique of cryptography is adopted to protect the data confidentiality and integrity. The typical examples include symmetric and asymmetric encryption algorithms. Nevertheless, since each algorithm has its own strength and weakness, how to apply these algorithms to wireless ad hoc networks is presently a debatable topic. Specifically, symmetric key algorithms are a typically efficient and fast cryptographic methodology, so it has significant applications in many realms. For a wireless ad hoc network with constraint computational resources, the cryptosystem based on symmetric key algorithms is extremely suitable for such an agile and dynamic environment, along with other security strategies.

In this section, we explore the issue of applying both symmetric and asymmetric key algorithms for data confidentiality. Then we propose an advisable solution for the application of these cryptographic algorithms, which not only takes advantage of symmetric

key to provide the effective protection of data transmission, but also employs asymmetric key to secure the admission of a node. On the other hand, we introduce the concept of selective encryption into the design of data protection mechanisms. Due to the characteristics of symmetric key algorithms, we present the principle of selective encryption and propose a probabilistically selective encryption algorithm based on symmetric key. Thus, we avoid the weakness of these two cryptosystems and make use of their advantages to establish a secure communication environment, by means of the combination of symmetric and asymmetric key algorithms.

5.1 Motivation

In an open or hostile environment, data confidentiality generally is considered to be the foundation of securing communication connections for a group of trustworthy communicating parities. Specifically, these communicating parities do not expect an unauthorized thirty party to access their exchanged information and even to tamper it. Therefore, data confidentiality and integrity becomes a prominent concern in the realm of information and network security. Particularly, for a dynamic and agile environment such as wireless and/or mobile networks, which is vulnerable to adversaries, data protection is significant in such a context. At present, although a number of security solutions have been developed, there is still a lot of work to carry out about how to effectively apply them to wireless ad hoc networks.

There are symmetric and asymmetric encryption algorithms existing in the area of cryptography. In symmetric key algorithms, a shared symmetric key (which is also referred to as secret key) is known by both the sender and the receiver(s) during the process of data transmission; whereas, a pair of keys is used by asymmetric key algorithms: the public key is widely distributed, but the private key is kept confidential. Figure 5.1 illustrates a secure connection between *Alice* and *Bob* with symmetric key encryption/decryption, even if there is an eavesdropper *Eve* who still cannot access the

content of communications without the appropriate key. Currently, a primary concern about these cryptographic algorithms is security extent and computational cost. Symmetric key algorithms have a lower computational cost associated with a lower security extent, when compared to asymmetric key algorithms which have a much heavier computational cost but with a higher security extent. Apparently, it is not efficient to employ asymmetric key encryption at all times in a wireless network, though the PKI provides sufficient security; however, symmetric key encryption cannot provide ideal security even with lower computational consumption. Thus, how to find a proper balance between symmetric and asymmetric key algorithms is crucial to our current research.

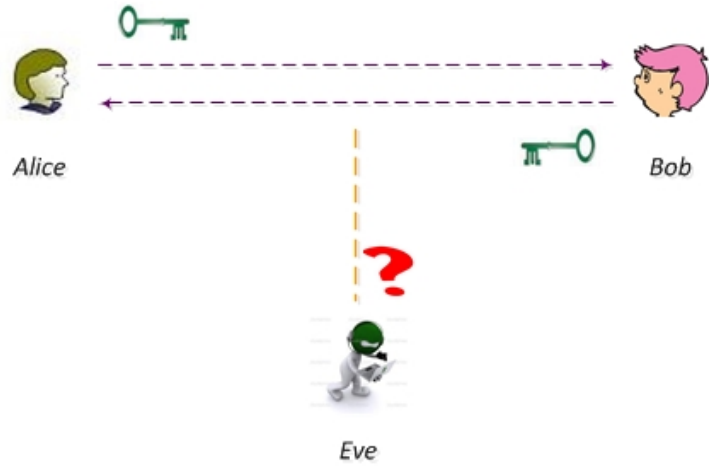


Figure 5.1: Data Protection between *Alice* and *Bob*

5.2 Related Work

With the development Internet and network security, a number of works have been carried out in the area of data protection [22, 65, 71]. In this section, we will review the related work about cryptography in Internet and wireless networks. As the mainstreams of data protection, cryptographic techniques act as the essential tools for data confidentiality and integrity. However, other methodologies are also employed to protect the confidentiality

of exchanged information. For instance, Zheng *et al.* [120] use the method of message segmentation to disseminate a message in multiple packets, so that these packets can be sent through a variety of intermediate nodes to avoid being intercepted by the same intermediate node. Consequently, the intermediate nodes which transferred the message cannot re-construct the original message just based on some of the intercepted segmentations. The hash function is also a kind of data protection approach as proposed by Huang and Medhi in [46]. During the process of key distribution, each group member will be issued a hash value and all nodes' hash values form a hash chain. Furthermore, the member can generate its own group key based on the issued hash value and then communicate with other members by using its group key. Maggi and Sisto [72] make an attempt at a new methodology of data protection through applying the mobile agent technique in distributed systems. Simultaneously, various cryptographic mechanisms can be involved into a mobile agent to secure a communication link.

5.2.1 Cryptography over Internet

Nowadays, in the realm of information security, a certificate authority is always deployed to issue the corresponding keys to each entity and also responsible for authentication. Bidder and Weiler [107] discuss the public key issues over Internet. One of most important purposes for PKI is to secure emails exchanged through Internet. However, how to securely distribute and to verify public key is an existing difficulty faced by PKI. OpenPGP can verify a public key by means of a combination of key certificates and those distributed public keys. Aiello *et al.* [1] design a rekeying protocol with a series of secure establishment messages. Totally, four messages are initiated to exchange the secret key and public key: 1) message 1 initiates the authentication based on an indication ID_R included in this message; 2) message 2 is the reply message from the responder with a signed copy of its exponential, a random nonce, and a secret HK_R acquired from an authenticator; 3) message 3 replies the message 2 to the responder and authenticator; 4) message 4 contains application-specific information and other identification information

for both sides of the communication. Additionally, the employment of password becomes a principal method to authenticate a user, by which the user can prove to servers his/her validation. Saito *et al.* [94] show the use of password-based authentication mode in SSL (Secure Sockets Layer) communications. Based on the discussion of two types of authentication methods (basic authentication and digest authentication), they point out that client authentication is more important to secure the SSL communications. Heikkila [43] investigates existing encryption tools and software especially designed for mobile devices, such as laptop, PDA, etc. The methods of encryption and password are combined to provide protection to the data in portable devices.

5.2.2 Symmetric Key Algorithm

In general, symmetric key encryption is regarded as a light-weighted algorithm for data confidentiality. Xu [116] discusses the traditional usage of symmetric key as group keys and a group key can be broadcast to all of its members with a certain limitation, such as timestamp, sequence number, etc. They also proposes an adversary model which exploits those out-of-date keys by compromising a legal group member, and accordingly presents countermeasures based on a rekeying mechanism. The Rijndael algorithm [50] is used to encrypt blocks of bit stream based on symmetric key. To extend from Data Encryption Standard (DES), Rijndael adds an initial round (AddRoundKey) in order to combine the key with shift state, followed by *XOR* operations with the matrix obtained from the former MixColumn phase. Aikawa *et al.* [2] describe a rotation-based encryption algorithm which is called MX. This proposed algorithm is similar to DES and takes advantage of two subkeys without lookup tables, in order to simplify the step of key schedule. For each rotation, the transformation of MX only makes changes about the parameters of rotation.

As an important application of symmetric key, it usually serves as the group key for the communications within a specified group of nodes. Such a group key is associated with a certain attributes, such as timestamp, access privilege, etc. Luo *et al.* [71] design

a key distribution scheme to allow a group of nodes to share a session key. During the initial phase of a group, the initiating node will generate a session key and then use an iterative approach to forward this key to the rest of the nodes in a number of iterative steps. The whole key pre-distribution scheme consists of three phases: 1) setup phase, which generates a key pool to provide the group key; 2) secure key share phase, which is a key pre-distribution process; 3) session key generation phase, which forms a session through the contribution from all members of a group. After the partition of pre-distributed key pool, any pair of nodes can share a certain amount of keys in order to establish a secure link between them.

Figure 5.2: Traditional Symmetric Key Update

The update of symmetric keys is usually accomplished through a central authority as well. Figure 5.2 illustrates a centralized key update and distribution procedure. The *CA* wants to distribute new session keys to *Alice* and *Bob* respectively, so it uses the corresponding old session keys SK_A and SK_B to respectively encrypt their new session keys, $SK_A^{'}$ and $SK_B^{'}$, and distribute them. Chan [23] proposes a distributed key pre-distribution scheme for MANETs. During the process of key distribution, a pair of node is required to select a subset of keys from a pre-generated key pool, and simultaneously, guarantees such a subset to own a common key which is not included by other nodes' key subsets. In addition to the distribution of group keys, other issues also attract

much attention, for example, how a new member is assigned a group key when it joins the group, how a member's group key is revoked when it leaves the group, and how to update a group key, and so on.

Sometimes, symmetric keys are used as a session key to manage a localized group. For example, Liao *et al.* [65] involve the latitude/longitude coordinates provided by GPS into message encryption, together with a session key. The session key will be updated periodically between the server and its mobile users. Pirzada and McDonald [87] present a Kerberos-related symmetric key service for MANETs. Multiple servers are deployed and secret keys are only shared between users and their corresponding server. If a user A wants to communicate with another user B, A must get authenticated from its corresponding server and then access B with a session key only known between user A and user B. During the process of mutual communication between A and B, their session key is issued with a timestamp, so that new authorization is needed once the session key expires. Damodaran *et al.* [33] present a secure key management framework for multicast communications in wireless and mobile networks. In their mobile architecture, a session key is used as the group key to manage a subgroup in which each member is authenticated before it is allowed to access the group key in this subgroup.

5.2.3 Asymmetric Key Algorithm

The issue of public key management becomes more difficult in a wireless ad hoc network due to the constraints on computational capability and the absence of centralized management. Steiner *et al.* [99] develop a key agreement protocol for dynamic groups which considers every aspect of group key operations, including key issue, key revocation, etc. The proposed key service CLIQUES uses the Diffie-Hellman (D-H) method as the basis of key exchange and consists of two primary operations, initial key agreement and auxiliary key agreement. Sjoholm *et al.* discuss the issue of group key agreement in [98]. Their scheme inherits the TGDH (tree-based group Diffie-Hellman) protocol, in that the keys own the characteristic of independence and guarantee the functionality of the

issued group key given that one or several points of failures. In the meantime, a group master key is used to generate a content encryption key (CEK) for message encryption and decryption. Figure 5.3 shows a public key encryption process between two users A and B.

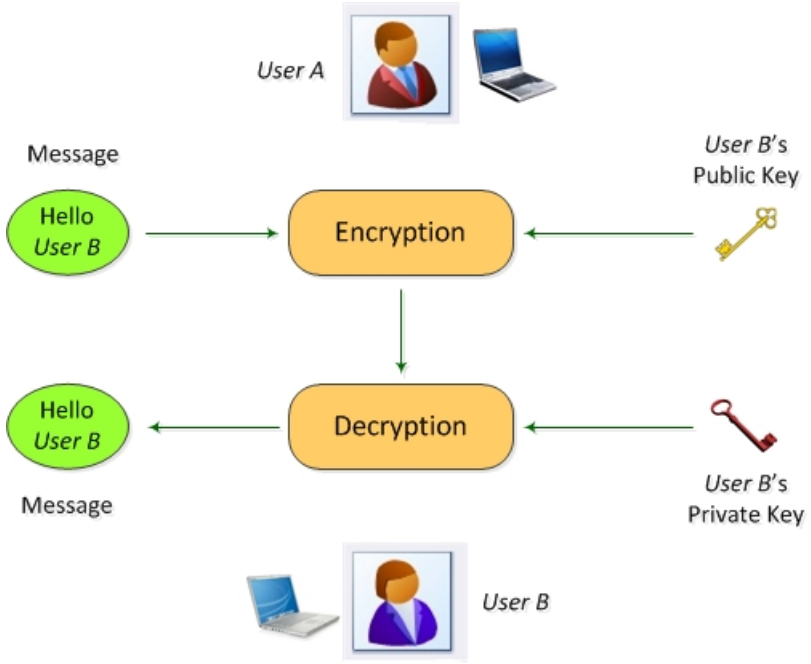


Figure 5.3: An Illustration of Public Key Encryption

Capkun *et al.* [22] propose a self-organized public key management solution, which avoids the use of central authority: all public keys and private keys are generated by individual nodes. In their system, authentication is achieved through a chain of public key certificates. Additionally, a node can revoke its certificate by *explicit* and *implicit* methods: *explicit* will revoke a certificate with a revocation statement, and *implicit* will revoke a certificate without any notification when the certificate expires. The concept of value-base utility (VBU) is adopted by Reyes *et al.* [93] to improve Capkun's work in terms of public key management. Because it is not always assumed that all nodes in a MANET are willing to cooperate for the request of public key certificates, selfish nodes will break the chain of certificates. VBU is regarded as a kind of incentives to measure

how a node satisfies with another node's cooperative requirement, in order to obtain the node's resource. Dahshan and Irvine [32] explore the issue of public key certificate following Capkun's work as well. The discovery of certificate chain is combined with route discovery, and any node along with the route will append its public key certificate in the route discovery message, so that the destination node D can obtain the whole chain of certificate and send it back to the source node S. If a node wants to revoke a certificate that it issued, it will broadcast such a message to all its directly trusted nodes.

Zhang *et al.* [119] study the challenge of avoiding certificate-based PKI and take advantage of the identifier of the target node. Thus, a mobile node can easily derive an ID-based public key from the already known nodes, instead of being authenticated by a trusted third party. Hao *et al.* [42] design a PKI-based scheme for fast authentication in MANETs, which does not use a centralized authority, but employs the technique of trusted computing. Each node at the bottom level will be equipped with a TPM (Trusted Platform Module) to provide fundamental cryptographic functions. In their proposed system, the certificate building and verifying processes combine a node's public key and its identity for the purpose of authentication. A self-organized key management scheme is proposed by Merwe *et al.* in [104], in which a subordinate public key is computed from a node's public key and then the node computes its corresponding private key. Subsequently, each node can generate a certificate based on the subordinate public key as their authentication proof. Khalili *et al.* [56] discuss the issues of key distribution using PKI in ad hoc networks, and thereby propose a flexible and efficient key distribution mechanism using threshold cryptography.

5.2.4 Selective Encryption Algorithm

Recent research has been extensively carried on the area of cryptography and data encryption [9, 44, 51]. A variety of cryptographic techniques such as symmetric key, asymmetric key, digital signature, are developed to provide secure data protection. For

example, Thamrin *et al.* [102] study the issue of random number generation used for generating pseudo random number in a cryptosystem. They make use of a pseudo RNG (hardware-based pseudorandom number generator) and a true RNG (true random number generator) to form a hybrid generator. Such a combinational mechanism can enhance the randomness and reliability of key generation. Bao and Deng [9] design a generic and fast cryptosystem based on symmetric key encryption. By means of the combination of block cipher and stream cipher, they take advantage of both of their advantages: a secret key is distributed by the protection of block cipher, but the communicated plaintext is encrypted by the stream cipher. Ksters and Tuengerthal [59] investigate the computational consumption used by symmetric key encryption. Especially, they present a symbolic criterion for key exchange protocols, and also the ciphertext encrypted by their tagged keys does not need to carry any additional information.

Currently, selective encryption algorithms are mainly applied in the field of secure multimedia communications, as the volume of multimedia data is huge to transmit and the cost will be overwhelmed if each packet is encrypted or decrypted. Lian *et al.* [64] present a video encryption scheme for Advanced Video Coding (AVC) codec. In their algorithm, only those sensitive data are chosen to be encrypted, such as residue data and motion vector. Specifically, the intra-prediction mode is encrypted according to context-based adaptive variable length coding. Jun *et al.* [51] propose a two-way selective encryption scheme for MPEG video transmission, in order to speed up the process of encryption, in which each frame is sliced to m slices, each slice is first implemented with *XOR* operation, and then the resultant slice is selectively encrypted by using a symmetric key. Massoudi *et al.* [75] define a series of evaluation standard for JPEG 2000 compressed image transfer, including encryption ratio, cryptographic security, compression friendliness, format compliance, and so on.

In particular, selective encryption algorithms are more preferable by wireless networks because they can save energy for wireless devices. Potdar *et al.* [88] compare different MPEG encryption algorithms from multiple points of view. In addition to security level

and string size, they particularly emphasize the encryption speed within a real-time communication environment, including on-demand video conferencing scenario.

(1) A Lightweight Fast Media Data Encryption Scheme

Xiao *et al.* [115] propose an application instance of selective algorithms and adopt a lightweight media data encryption method. They employ traditional block cipher to encrypt the plaintext partially (part I), and then use the plaintext to encrypt the rest part (part II). By the modification of the ratio between parts I and II, the encryption speed is adjusted accordingly. Their algorithm is applied to video conference for wireless terminals. Specifically, in the basic algorithm, the plaintext is first divided into m segments with a same length. Then wireless terminals use traditional block cipher algorithm to encrypt one segment. Eventually, for the rest of *m-1* segments, the initiating terminal uses the plaintext of the previous segment as its stream cipher key, to encrypt the rest of segments.

(2) Hofbauer and Uhl's work

Hofbauer and Uhl [44] attempt to solve the issue of scalability and security in bitstream transmission. Their solution is selective encryption where only a part of the content is encrypted, enough to ensure security, but at the same time, little enough to keep scalability intact. When the bisstream is chosen to be encrypted, some important data, such as I frame, temporal or spatial information, has higher priority for encryption. Based on different Digital Rights Management (DRM) scenarios, they use a variety of methods to selectively encrypt the bitstream of MC-EZBC, such as confidentiality encryption, sufficient encryption and transparent encryption.

- Confidentiality Encryption: means complete security, except for the information the sender wants to give away.
- Sufficient Encryption: means it is not necessary to require full security, just enough

security to prevent intruding of the data.

- Transparent Encryption: means video receivers are able to see a preview version of the video but in a lower quality, while prevent them from seeing a full version.

5.3 A Hybrid Cryptosystem with Symmetric and Asymmetric Keys

After a community is established, data confidentiality is immediately considered to be the foundation of securing the communication among the nodes in this community. Especially, any unauthorized thirty party is not anticipated to access the exchanged information between communicating parties and even to tamper it. As we described before, cryptography is not only an indispensible component to build up a secure system, but also a fundamental tool to protect the exchanged information. Without encryption/decryption, anyone can obtain the communicating data and even easily jeopardize it. As such, a security system should enable each of its users only to access his/her corresponding granted rights. Therefore, in this section, we mainly concentrate on the issues of encryption and authentication, and aim to apply the combination of both the symmetric and asymmetric key algorithms to wireless and mobile networks. In this way, they not only provide enough security protection to data confidentiality and the security of a network, but also spend a reasonable cost on data encryption. Based on the special requirements of a wireless ad hoc network to the mechanism of data protection, we propose a hybrid application of symmetric and asymmetric algorithms.

5.3.1 Symmetric Key and Asymmetric Key Algorithms

By means of cryptography, an unhidden message (plaintext) is converted to an unreadable form (ciphertext), so that only the receiver can convert it back to the original state by pre-knowledge of a key. The process of message conversion is respectively achieved at

both sides of communicating parties [116]. According to the classification of existing cryptographic algorithms, we compare the strength and weakness between symmetric and asymmetric key algorithms in Section 5.3.2.

For a symmetric key algorithm, it allows the sender and receiver to know a shared key in advance, and encrypts/decrypts a message using the same shared key. Obviously, such a key is symmetric for both sides of the communicating parties. Nevertheless, since the key needs to be pre-distributed, a secure distribution approach is indispensible, and so key management is important for symmetric key algorithms. In comparison, an asymmetric key algorithm makes use of a pair of keys (public key and private key) which are different. Meanwhile, the public key is allowed to be distributed widely, while the private key is only known by the receiver instead of the sender. Theoretically, the lengths of public and private keys are long enough so that one key cannot be mathematically calculated from the other key. When encrypting a message, the message can be encrypted by the public key, but only decrypted by the corresponding private key. Therefore, asymmetric key algorithms separate the functions of encryption and decryption by using a pair of asymmetric keys.

5.3.2 A Comparison of Symmetric Key and Asymmetric Key Encryption

Currently, a primary concern about these cryptographic algorithms is how to find a right balance between symmetric key and asymmetric key algorithms. As we described above, they have different security extents and computational costs. According to the features of symmetric and asymmetric key algorithms, evidently they are totally different cryptosystems. Thus, we are interested in the following questions:

- What are advantages and disadvantages of these two categories of algorithms?
- Which algorithm is more suitable for a wireless ad hoc network?

Table 5.1 simply compares the differences between these two algorithms. Through this table, we can clearly see that it is difficult to judge which algorithm is better or which one outperforms the other, because each of them has its own pros and cons. As a comparison, symmetric key algorithms have lower computational cost and faster computational speed, associated with a lower level security; however, asymmetric key algorithms are more secure with a relatively higher computational consumption. Here, it is significant to find a tradeoff between these two methodologies. When applying them in a wireless ad hoc network, we propose the following principle: we adopt a hybrid cryptosystem which employs both symmetric and asymmetric key algorithms. Especially, the symmetric key is preferred with periodical key update at the stage of data transmission; whereas, asymmetric key is only used in some important stages, such as authentication, key distribution, rekeying, etc. As a result, we avoid the weakness of asymmetric key algorithms and use the advantages of symmetric key algorithms.

Table 5.1: A Comparison of Symmetric and Asymmetric Algorithms

Algorithms	Examples	Security	Cost	Time
Symmetric Key	DES	Low	Low	Fast
Asymmetric Key	RSA	High	High	Slow

5.3.3 Overview of a Hybrid Cryptosystem

Symmetric key algorithms provide good security protection with less computational cost, as well as less computational time, so such cryptosystems are efficient. When it comes to asymmetric key algorithms, though they provide better security for data confidentiality, they are much slower than symmetric key algorithms, and also consume more computational resource. Due to their inefficiency, asymmetric/public key cryptosystems, for example RSA, are unsuitable for wireless ad hoc networks where there are constraints on computing ability and energy. In fact, symmetric key systems, like DES, are still the

major tools for communication privacy and data authenticity in most networks [2].

Based on the SeDi community model, we intend to apply the combination of both symmetric and asymmetric key algorithms to wireless ad hoc networks as follows:

- Asymmetric key encryption will be applied to some important scenarios;
- Symmetric key encryption will be applied to the large scale or official data transfer.

The above arrangement is determined by the features of these two encryption algorithms, as well as the characteristics of wireless devices. Since the high computational consumption of asymmetric key algorithm and the limited resource of a wireless device, it is not practical for the wireless device to encrypt all its messages in asymmetric key algorithms. Thus, after a pair of nodes is authenticated and their data transfer medium is established securely, all official data transmission is protected by using symmetric key algorithms. As such, in order to enhance the security of the network, symmetric key encryption cannot fully carry on the protection of some important scenarios, such as, key distribution, authentication, etc. Thereby, asymmetric key encryption is employed in such scenarios. For example, in the process of key distribution, the public key of a newly joined node will be used to encrypt its assigned secret key by the central node and then distributed to the newly joined node. After the newly joined node receives the assigned secret key which is encrypted by its public key, it can retrieve the secret key by its own private key. Thus, the procedure of key distribution is secured. Through applying the combination of symmetric and asymmetric keys, it is suitable to dynamic and open environments. Eventually, these nodes can initiate their official data communication phase, and the communicating data is encrypted and decrypted by using the assigned secret key.

After a new node is accepted as the member of the intended group, it indicates that the newly joined node has been authenticated. Subsequently, any node of the communicating sides can initiate the data transfer phase by using the issued unique secret key, and encrypt the data which the sender wants to transfer to the receiver. Once

the receiver gets the transferred data, it will decrypt it using the shared secret key. As we described above, the public key of each member guarantees the secure distribution of each member's unique secret key, as well as their group key.

5.4 A Probabilistic Selective Encryption Scheme

In this section, we study the issue of selective encryption for wireless ad hoc networks. Then, we present a probabilistic selective encryption algorithm based on various security strategies, which encrypts the transmitted packets by means of probabilistic function and stochastically selective algorithm. By introducing probabilistic methodology and stochastic algorithm, a sender includes proper uncertainty in the process of message encryption, so that only entrusted receiver can decrypt the ciphertext and other unauthorized nodes have no knowledge of the transmitted messages on the whole. In addition, we also employ other security mechanisms to enhance the security of our proposed scheme.

5.4.1 Initial Consideration

A fundamental method of data protection in the area of information and network security is cryptography, which has been widely accepted as a traditional platform of data protection for decades. Through the data encryption and decryption, the protection of data confidentiality and integrity are achieved. However, based on the features of wireless devices, a wireless ad hoc network has special security and efficiency requirements for conventional cryptographic algorithms.

As we stated in Section 5.3, since wireless devices are usually equipped with batteries as their power supply, they have limited computational capability and the issue of energy saving is one of the most important concerns. Hence, an efficient selective encryption algorithm is a potential solution to save considerable power for wireless devices, and at the same time, to provide sufficient protection for data communication.

Through applying the selective and probabilistic methods, our proposed scheme en-

hances the reliability of selective algorithms, and avoids the relevance between different messages encrypted by symmetric keys. Thus, it effectively prevents data disclosure to untrustworthy nodes and economizes the overhead spent on the data protection for a network. Such a probabilistic solution is suitable to dynamic and open environments. For a wireless ad hoc network,

5.4.2 The Issues of Selective Encryption Algorithms

As we stated above, selective encryption are widely accepted in energy-aware contexts, due to the fact that they can reduce the overhead spent on data encryption/decryption, and improve the efficiency of the network. In this section, we present the principle of selective encryption and then study one of the most important methodologies.

The Theory of Selective Encryption

The purpose of selective encryption algorithms is to just encrypt a certain portions of the messages with less overhead consumption, but simultaneously, sufficient messages are encrypted to provide reliable safety to secure the transmitted message confidentiality. Through selective encryption, not all messages are necessary to be encrypted while the entire data transmission can be viewed to be secure on the whole. Selective encryption is able to improve the scalability of data transmission and reduces the processing time.

In the theory of selective encryption algorithms, uncertainty is involved in the message encryption process to determine the uncertain pattern of encrypted messages. Here, uncertainty can enhance the security of data transmission, as all messages are assumed to own equal importance. Thus, uncertainty becomes one of the paramount factors when designing a selective-based cryptosystem. Usually, the more uncertainty is involved, the more effective the cryptosystem is. Nevertheless, we also note that an efficient algorithm will reduce the complexity of selective encryption. Figure 5.4 is a schematic diagram of a selective encryption process.

Nowadays, selective encryption algorithms are primarily applied in the realms of

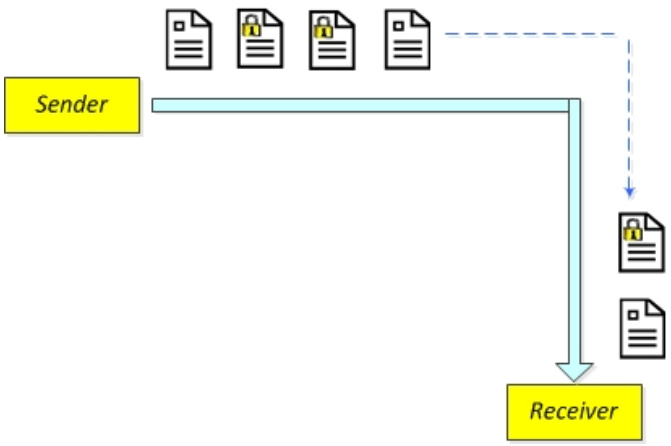


Figure 5.4: The Schematic Diagram of Selective Encryption

energy-aware environments or large-scale data transmission, such as, multimedia communications, mobile ad hoc networks, wireless sensor networks, etc. For multimedia communications, it often requires real-time data transmission, so tremendous audio and video data need to be transferred securely. Given that all multimedia data are encrypted, this will consume a great deal of overhead, so that multimedia data is difficult to transmit timely and the quality of communication cannot be guaranteed. As such, in a WSN, each device uses battery as its power supply and thereby has constrained computational ability, so a sensor cannot spend too much computational cost on data encryption and decryption. Under such circumstances, the design of a selective encryption algorithm with less processing time but with relatively high security level is extremely significant.

Probabilistically Selective Encryption

With the definition of selective encryption, we first realize during the data communication process, it is not necessary to encrypt all messages. When the communicating data is scalable or a network is aware of its limited computational resource, it is not really needed to provide full security protection on all exchanged information. On the other hand, we do not wish that eavesdroppers are able to reconstruct the contents even if they can intercept partial or full transmitted messages.

As a major selective encryption methodology, the probabilistic method provides sufficient uncertainty to a selective-based cryptosystem. In order to adapt the scalability and to improve the processing capability of a cryptosystem, the cryptosystem just partially encrypts the transmitted messages that it wants to protect based on a certain probability. In the meantime, probability is involved in the procedure of selective encryption, so that those selected messages are encrypted in a deterministically random way. By means of the inclusion of probability, nobody will exactly know which messages are encrypted except the communicating parties. Thus, even if there are malicious attackers which are able to intercept the communicating messages, they still cannot fully obtain the selectively encrypted messages or reconstruct all messages. Consequently, the probabilistic method is widely employed to enhance the confidentiality of the communications.

5.4.3 The Overview of our Proposed Selective Encryption Algorithms

In this section, we will present the design of a probabilistic selective encryption algorithm step by step, which not only reflects the idea of probabilistic encryption, but also uses both of symmetric key and asymmetric key. Specifically, our proposed algorithm aims to involve sufficient uncertainty into the encryption process, while providing satisfactory security protection to communicating nodes. In the ad hoc network we discuss, the links between wireless nodes are always bi-directional and every wireless node has enough computational power to finish these operations.

Secure Key Distribution

In order to protect the confidentiality of communicated messages, our proposed selective encryption algorithm takes advantage of major categories of cryptographic techniques, symmetric and asymmetric key algorithms, to guarantee the security of exchanged information. In this way, all official data communication between two nodes will be encrypted

through symmetric key, and in the meantime, these symmetric keys will be distributed by public key encryption algorithm.

In a network, when a node wants to communicate with another node, a secret key (symmetric key) will be generated for their communication. Let us denote the initiating node as S and receiving node as R. If an initiating node S moves into the neighborhood of node R, it will inform the node R of its public key for the authentication between them. The receiving node R then assigns a secret key to the initiating node S for the purpose of encryption/decryption. In order to distribute the secret key securely, R will encrypt this secret key using the public key of node S before sending it. Furthermore, R generates different secret keys for different initiating nodes. Thus, each sender has a unique secret key for communicating with the receiver and all information is encrypted using the corresponding secret key.

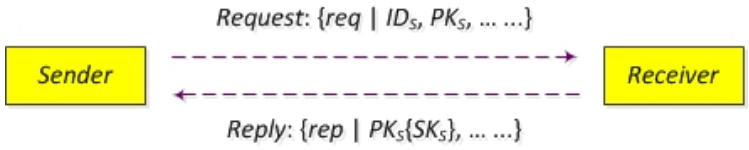


Figure 5.5: The Schematic Diagram of Symmetric Key Distribution

The above Figure 5.5 illustrates the procedure of secret key distribution between a pair of nodes. The message's sender composes a communicating request message *req* which contains not only its identifier ID_S, but also its public key PK_S, for the purpose of their later mutual authentication. Once the receiver gets such a communication request, a secret key (symmetric key) SK_S will be generated by the receiver and encrypted using the public key PK_S of the requester, which is included in the communicating request message. Later, the receiver composes a communicating reply *rep* message and replies it to the communicating sender, in order to indicate that their communication has been successfully established. After the sender obtains the response from the receiver, it will use its corresponding private key PR_S to decrypt the secret key SK_S issued from the receiver.

A Toss-A-Coin Selective Encryption Algorithm

First of all, in order to provide sufficient security to data encryption, in the first proposed approach, we choose a relatively high proportion as encryption ratio. Since the toss-a-coin algorithm is a basic approach, little uncertainty is involved. For all transmitted messages, we divide them to two groups: the odd number messages and the even number messages. For instance, messages $M_1, M_3, M_5, \ldots, M_{(2n-1)}$ represent the odd number messages; messages; $M_2, M_4, M_6, \ldots, M_{(2n)}$ represent the even number messages. When the sender needs to decide which group should be encrypted, it makes use of a toss-a-coin method to determine whether the even number messages or odd number messages are encrypted.

As an example, we consider the following scenario, in which the even number messages are encrypted. After the method of toss-a-coin is applied, the sender makes the decision that only the even number messages $M_2, M_4, \ldots, M_{(2n)}$ are encrypted. Thus, half of the whole messages are chosen to be encrypted and this approach shows a basic selective encryption algorithm with a semi-determined encryption pattern. As we described before, the more data are encrypted, the more secure the communication is, but the more overhead is spent. Hence, the value of encryption ratio here is tentatively determined to be 0.5, which means that 50 percents of the communicated data will be encrypted.

A Probablistic Selective Encryption Algorithm

Here, we propose a probabilistically selective encryption algorithm, which uses the advantages of the probabilistic methodology, aiming to obtain sufficient uncertainty. During the process of sending messages, the sender will randomly generate a value to indicate the encryption percentage, which represents how many messages will be encrypted among the transmitted messages. Then, the sender uses a probabilistic function to choose the already deterministic amount of messages to encrypt them. Figure 5.6 shows the flow chart of the probabilistic selective encryption. We can see that more uncertainty is included

to the probabilistic encryption algorithm, in comparison to the toss-a-coin approach, since the encryption ratio is randomly decided and the encryption pattern is not pre-determined. Moreover, this proposed selective algorithm is comprised of the following three phases:

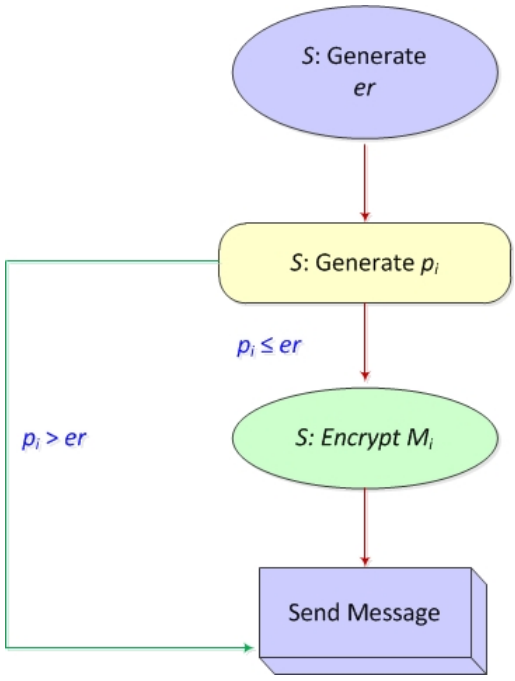


Figure 5.6: The Flow Chart of Probablity Selective Encryption Algorithm

(1) The sender of communciating parties S will first apply a random generator RNG to randomly obtain an encryption ratio er, which determines the percentages of encrypted messages among all messages. Here, in order to ensure that enough data are able to be encrypted so as to provide sufficient security protection, the generated encryption ratio should be higher than a pre-determined value of security requirement SR (SR means that data communication is secure if there are SR or more percents of messages are encrypted).

$$S \xrightarrow{RNG} er \mid \{er \geq SR\} \tag{5.1}$$

(2) Then the sender S will employ a probablistic function PF to generate an encryption probability p_i to determine if one message M_i will be encrypted or not.

$$S \xrightarrow{PF(M_i)} p_i \tag{5.2}$$

(3) Eventually, the sender selects the messages to encrypt based on the above predetermined encryption ratio er. For example, once S finds out that the encryption probability p_i is less than or equal to the encryption ratio er, it will encrypt the message M_i using its secret key SK; otherwise, this message will not be encrypted accordingly.

$$\begin{cases} S \rightarrow SK[M_i] & p_i \leq er \\ S \rightarrow M_i & p_i > er \end{cases} \tag{5.3}$$

Thus, the probablistic selective encryption algorithm integrates both the probablistic method and stochastic strategy, in order to increase the uncertainty in the process of message selection. As we discussed in the theory of selective encryption, the more uncertain the encrytion algorithm is, the secure data communication is, based on the assumption that sufficinet data is encrypted to provide reliable security.

The Exchange of Selective Pattern

Once the sender of the communicating parties transfers the official traffic by means of selective encryption, it will let the corresponding receiver know the encryption pattern of the exchanged messages in a secure way. First, the sender will summarize the pattern based on those selectively encrypted messages and indicate which messages have been encrypted. Subsequently, it composes the encryption pattern in a pattern-related message and then sends it to the messages' receiver. In order to securely distribute this pattern message, the public key of the receiver is also used to encrypt this pattern-related

message. Thus, the receiver can use its corresponding private key to decrypt the pattern message and thereby keep track of the information of encrypted messages. Through such a public key based method, the process of pattern information exchange is kept confidential only to the communicating parties.

5.5 Security and Performance Analysis

5.5.1 Security Analysis

In this section, we summarize the security features specified below, and show that the hybrid cryptosystem of symmetric and asymmetric key algorithms which we proposed in Section 5.3 is able to provide sufficient security protection for the data confidentiality and authentication process [21, 82].

Theorem 1. *Public key is able to protect the process of key distribution.*

Proof. Based on the asymmetric features of public key algorithm, our proposed cryptosystem can securely distribute the secret key issued by the central node to its corresponding members. First of all, the private key PR_N of a node N is kept secret for that node, so that other nodes M cannot decipher the intercepted key distribution message. When the central node C distributes the unique secret key $SK_{(C,N)}$, it will use the public key PK_N of N to encrypt this SK first and then sends it. Especially, a member's public key and private key is unique to this member, the pair of asymmetric key can authenticate the identity of the member from the central node point of view. Therefore, using a public key as the tool of key distribution, the key exchange process between the central node and its member can be protected securely.

Lemma 1. *The mechanism of one-node-one-key prevents the disclosure of the issued secret key to unwanted third party.*

Proof. A major concern about applying a symmetric key as the group key is that multiple members in a group share the same secret key, which leads to actually unqualified members may still access the shared secrecy, so as to decrypt the ciphered messages. For

instance, when the central node C wants to communicate with N_i, it has to use the shared group key GK to encrypt their communicating message $MSG_{(C,N_i)}$. Thus, other members N_j can also receive this message and use the same key GK to decrypt the message. However, the $MSG_{(C,N_i)}$ is not related to N_j at all, so we can see that the group key sometimes enables the exchanged information to be exposed to unwanted nodes.

In our cryptosystem, each node N is assigned a unique secret key $SK_{(C,N)}$ only for its communications with the central node C. Other members will not decipher the encrypted message, even if they may intercept it. Typically, even if there is a certain member N_X in the existing group C is compromised and the secret key $SK_{(C,N_X)}$ is revealed to malicious node M, M can access the relevant information only about the compromised N_X, but the compromised member N_X still cannot eavesdrop on other nodes' communications. This is due to the fact that the keys encrypted for other nodes' messages are different from the compromised nodes key. Thus, it totally avoids the risks from the shared group key. Compared to the situation where multiple nodes share a group key, our mechanism is more resilient to the disclosure of secret keys.

Lemma 2. *Periodical key update improves the security of symmetric key.*

Proof. Another concern about symmetric key is given rise to by the fact if a secret key is used for a long time, then this increases the chances of being exploited by malicious nodes. Here, it is possible that some malicious nodes exist in the network to make cryptanalysis and exploit those stale keys to disturb the network. One of potential solutions is to periodically update the secret keys and keeps these keys as fresh as possible, which will greatly relieve the above situation. Assumed that malicious nodes eavesdrop on some stale secret keys but intercept the messages encrypted by a newly update key, malicious cryptanalysis will not work effectively due to the change of secret keys. In addition, as we stated before, one major disadvantage of symmetric key algorithms is the insufficient security extent. Nevertheless, the periodical update of symmetric keys will greatly improve the security of symmetric key algorithms, since it will not provide

enough time to an attacker and the keys have been updated so that the attacker cannot corrupt the keys timely.

Theorem 2. *Symmetric key can protect the data transfer phase with the timely periodical key update.*

Proof. For symmetric key algorithms, the concerns about the security arise from the length of symmetric keys, malicious cryptanalysis may take place if an attacker spends enough time on the cryptoanalysis of these keys. Definitely, if a secret key has a short length, it will cost relatively less time for the attacker to compromise the key. Whereas, supposed that the secret key has been updated before it is compromised by the attacker, then the attacker cannot exploit the weakness of symmetric keys and sufficient security is guaranteed by symmetric keys. At the same time, the concern of the shared nature of symmetric key can be mitigated due to our one-node-one-key mechanism, as the shared parties of the same key have been reduced as little as possible. As a result, these strategies effectively solve the issue caused by the short key length and provide durable protection to the users of symmetric key algorithms.

Lemma 3. *The mechanism of the unique secret key provides reliable protection for authentication.*

Proof. In a localized group, the central node C will issue a unique secret key $SK_{(C,N)}$ to each of its members N. Thereby, each member's secret key will be different and there is not any member can decipher other members' messages. For example, node N_i will have a distinctive secret key $SK_{(C,N_i)}$ with node N_j ($SK_{(C,N_j)}$), so that N_i cannot decrypt the encrypted messages $SK_{(C,N_j)}[MSG]$ of N_j, *vice versa*. In particular, when a secret key becomes a unique evidence for a specified entity, the unique secret key also acts as a means of authentication. Thus, the unique secret key not only becomes a proof of an identity for each member in the corresponding group, but also achieves the authorization function to granted members, because only the sender and receiver can decrypt the encrypted messages with the unique secret key. After the central node issues the unique secret key, both sides of the communicating parties (the central node and this

member) can authenticate each other using this unique secret key. They can consider their communication to be mutually authenticated, not only from the central node point of view, but also from the member point of view.

Lemma 4. *The public key and private key provide additional protection for authentication.*

Proof. In addition to the unique secret key, the public key and private key also play an important role during the process of authentication. From the side of the group member N, its private key PR_N is confidential in the entire group. Without PR_N, any other node cannot obtain the content encrypted by PK_N. When N sends its membership request to the central node C, C will issue a unique secret key $SK_{(C,N)}$ to N. Especially, the unique secret key will be distributed with the protection of PK_N. Once the newly joined member N uses its private key PR_N to obtain its membership issued by the central node C, it will deem that the central node has been authenticated, from the member point of view. Hence, the public key and private key can prove the successful authentication of the central node.

Theorem 3. *The node authentication procedure is protected with the public key and the unique secret key respectively.*

Proof. Mutual authentication is able to be achieved with the protection of the public and unique secret keys. As we described in Lemma 3, the unique secret key is equivalent to a confidential proof for the newly joined member, so it can authenticate the member as an authorization method. As such, as we stated in Lemma 4, the public key and private key are unique for the member as well, the newly joining member can know whether the central node agree to issue membership securely and if the central node is authenticated through the pair of keys. To sum up, the combination of the unique secret key and public key can authenticate the both communicating parties well.

5.5.2 Performance Evaluation

In this section, in order to study our proposed probabilistic selective encryption scheme and observe its characteristics, we carried out some experiments within a wireless ad hoc environment based on Network Simulator *ns2* [49]. In the setup of our experimental environment, 30 nodes are deployed and the transmission range of each node is set to $250m$ without fading effect. During the process of communications, the traffic is generated over UDP. Each experiment is run for $4000s$ of simulation time. The standard DES algorithm is employed for communication between a pair of nodes and the secret keys have an effective 56-bit key (totally 64-bit). During the simulation experiments described from this point onward, these compared systems are all run under the identical scenario.

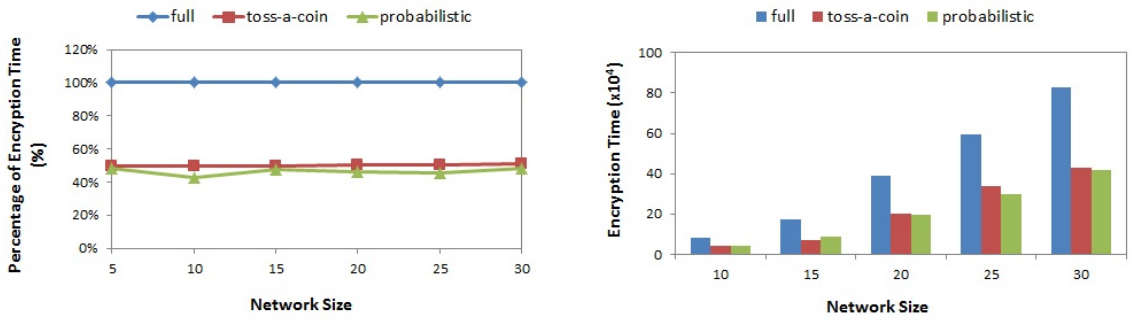
As we stated before, we will utilize two approaches as the comparable models to our proposed probabilistic approaches. The first approach will encrypt all messages without any selective encryption, and the second approach is the toss-a-coin approach. For the purpose of simplification, we use "full" to represent the first approach, "toss-a-coin" to represent the second approach, and "probabilistic" to represent our proposed probabilistic approach. The performance metrics for evaluating the probabilistic system are listed below:

(1) *Encryption Proportion*: the ratio of the encrypted messages to the messages that are not encrypted;

(2) *Encryption Time Percentage*: the percentages of the total time spent on encryption and decryption of selected messages to the total time encrypted all messages;

(3) *Encryption Time*: the overall time is spent on the message encryption and decryption;

(4) *Saving Time*: the ratio of the saved time from encryption to the overall processing time.



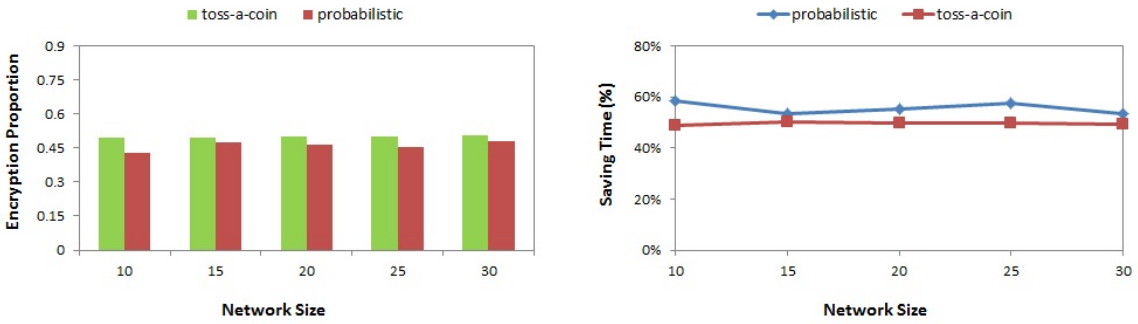
(a) (b)

Figure 5.7: Encryption Time *vs.* Network Size

First of all, we compare the performance and efficiency of these approaches. Figure 5.7 illustrate the comparison of encryption percentages and time based on three approaches. In Figure 5.7 (a), we can learn that both toss-a-coin and probabilistic have an obvious lower encryption time percentage than full encryption, which is caused due to the fact that selective encryption takes effect and the processing time is greatly reduced. In Figure 5.7 (b), the time spent on encryption/decryption is compared to show that full encryption takes a longer time than both toss-a-coin and probabilistic encryption. This means that data transmission can be speeded up by virtue of toss-a-coin and probabilistic encryption. Hence, selective encryption is more efficient than full encryption and it is able to better save the computational resource of a wireless device.

Figure 5.8 compare the overhead spent on toss-a-coin and probabilistic encryption respectively, based on their encryption efficiency and effects. Figure 5.8 (a) demonstrates that probabilistic encryption has a little lower encryption proportion than toss-a-coin encryption but it is more flexible than toss-a-coin encryption. Because probabilistic encryption does not fix the encryption probability, the encryption proportion fluctuates in a relatively larger range. Thus, probabilistic encryption owns more uncertainty than toss-a-coin encryption, which matches with our expectation. In Figure 5.8 (b), the comparison focuses on their efficiency and the factor of saving time is taken into account. Probabilistic encryption has a higher saving time when compared to toss-a-coin

encryption, indicating that it has a more fluctuated pattern and spends less time on encryption/decryption. Therefore, through the comparison of their efficiency, we learn that the selective encryption does help reducing the encryption overhead and improving the efficiency of encryption.



(a) (b)

Figure 5.8: Encryption Proportion *vs.* Network Size

5.6 Summary

Data confidentiality is an important topic in wireless ad hoc networks at all time. In this chapter, we propose a hybrid cryptosystem through applying symmetric and asymmetric key algorithms to provide reliable and secure protection to data confidentiality in a wireless environment. Our system can sufficiently make use of the advantages of both these cryptographic algorithms and avoid their weakness. On the other hand, we have presented a novel solution for selective encryption to achieve data protection effectively while with reasonably costs. The probabilistic and stochastic techniques in our proposed solution guarantee the security for data communications between the messages' sender and receiver. The factor of encryption probability involves the uncertainty to data encryption. Thus, a network not only obtains enough protection in terms of security, but also improves the efficiency of data encryption.

Chapter 6

ToA: A Token-based Authentication Scheme

Authentication is essential for secure network management especially in wireless networks, since it is able to identify any unauthorized network access and thereby to decline such requests. In the Internet or wired networks, a node's authentication is achieved through a *triple-A* framework (authentication, authorization and accountability). Nevertheless, such a framework usually needs centralized management which is not suitable for wireless networks, due to the absence of pre-deployed infrastructures. Presently, an important concern is how to manage a wireless network resource in a distributed way, and thereupon, to securely authenticate a legitimate node to access correct information or communicate with those authorized nodes. Thus, it is significant to apply traditional authentication techniques in wireless ad hoc networks, in order to enhance the reliability of authentication. To meet the design objectives, we propose a two-factor authentication algorithm to deal with the issue of authenticating nodes based on the concept of token-based authentication.

6.1 Introduction

Nowadays, a number of security strategies have been proposed to secure a network, and authentication becomes one of the most important security components for the network. In the traditional security models, authentication, authorization and accountability (AAA) are usually combined together to provide reliable identification verification services. For instance, authentication can validate an entity's identity; authorization enables an authority to grant a certain rights or resources to some specific entities, after they are authenticated; accountability will check up the records of the entity to its behavior, to detect if there is any security violation. Figure 6.1 illustrates the relationship among these methods. In general, a certificate authority will be responsible for the management of authentication, authorization, and accountability. Specifically, when a user sends a request to the authority, the authority will ask a credential from the user. After presenting the required credential, the authority will compare it to an access control list (ACL), to determine whether to grant or deny the presented request. When the grant of an access is declined based on the access control list, the user cannot access the required resource or execute the wanted operation. If there is a match between the credential and the ACL, then the user is able to operate in an authenticated way.

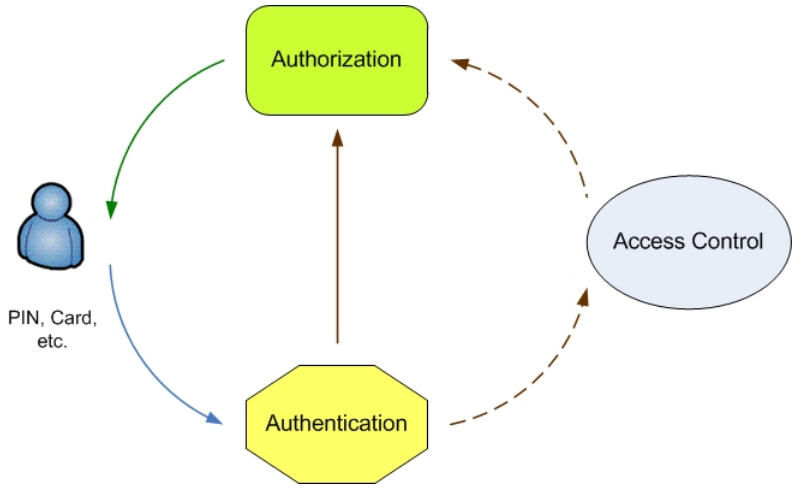


Figure 6.1: The Relationship of *Triple-A* Methods

However, in some wireless ad hoc environments, it is evidently impractical to deploy the infrastructures to act as central authorities because of the characteristics of wireless devices and the special requirements of such networks. Without centralized management, the traditional authentication management framework has to be modified to adapt the flexibility of wireless ad hoc networks, and even new authentication strategies are desired by wireless ad hoc networks. On the other hand, data protection is indispensable for an open communication environment, since data transmission channels are shared between wireless devices. At present, the asymmetric cryptographic key is considered to be a promising means of authentication, as it is not only the major protection method for data confidentiality, but also it can provide reliable authentication and avoid the involvement of other redundant authentication mechanisms. If such security mechanisms can be applied to dynamic and vulnerable context, such as in a wireless ad hoc network, they can achieve the distributed authentication, and at the same time, reduce the overhead spent on authentication.

As we described before, digital credentials are difficult to be managed in a centralized approach in a dynamic or agile environment. Here, how to authenticate an unknown node reliablly is one of our current research directions. We mainly concentrate on the following issues: 1) how to achieve a node's authentication in a secure and distributed method; 2) how to apply the existing symmetric or asymmetric key algorithms for node authentication in wireless environments. In the light of the features of the above cryptographic algorithms, we propose a two-factor authentication scheme, which employs the theory of two-factor authentication and is based on asymmetric key algorithms. Additionally, symmetric key algorithms are also applied to secure the communicated data in an efficient way. Our token-based authentication approach eliminates the weakness of single-factor authentication, and enhances the reliability of authentication. Hence, only when a network has a reliable authentication mechanism, it is able to ensure that the network includes trustworthy nodes.

6.2 Related Work

Authentication can prevent unwanted nodes in a wireless network from accessing the resources or data that are not granted to them. Some works have been done to provide a variety of authentication mechanisms and enhance the security in a wireless and mobile environment [47, 55, 66]. The earlier applications of authentication are developed for electronic commerce and Internet. Wang *et al.* [110] study the issue of authentication security and present a set of principles "authentication set" to analyze the security of the authentication schemes. This authentication set defines some new concepts, such as "cover authentication set", "attack authentication set", "watermark-based authentication set", etc. These definitions clearly classify the authentication schemes according to their ability to prevent attacks and security functionalities. Chang and Shin [25] develop a distributed authentication protocol which enables nodes to authenticate another communicating party through its identity. In addition, the identity of a server which is responsible for authenticating nodes also needs to be proved to those nodes. Thus, mutual authentication can make both parties of communication authenticate each other, in case that individual nodes are connected to malicious servers.

6.2.1 One-factor Authentication

Password is a traditional but widely accepted method for authentication, since a user is able to provide a personal identification number (PIN) at the initial authentication establishment stage and keep it confidential. At all times, password is regarded as a classic instance of one-factor authentication. Currently, a variety of factors are introduced to improve the reliability of one-factor authentication and different techniques are employed to enrich the possibilities of authentication factors. Liao *et al.* [66] present a password authentication scheme which is comprised of three phases: registration phase, login phase, and authentication phase. In the login phase, a user will use his/her smart card, keys of IDs and/or passwords to login. The smart card terminal composes a request

message to remote server to get the input information verified. Fan and Lin [37] employ the technique of biometrics to design highly security remote authentication systems. In particular, a three-factor authentication solution is proposed to combine biometrics, smart card and password. For a user's biometrics, their proposed scheme allows the central server to perform its own authentication to biometrics secretly. The biometrics information is only kept secret in the central server and even is not allowed to be released to other remote servers. Additionally, cryptographic keys sometimes are used as authentication factors as well. Tartary and Wang [101] investigate the multicast stream authentication problem and take advantage of both hash function and MD5 coding for authentication. In order to reduce packet overhead and improve computing efficiency, hash chains are adopted by the packets' receiver to check the validity of received elements upon reception.

ZCK: a Zero Common-Knowledge Authentication Protocol

Weimerskirch and Westhoff [114] present an efficient zero common-knowledge (ZCK) authentication protocol with symmetric key and identification, which establishes a trust relationship over nodes with moderate power supply through a mutual signature verification approach. In the meantime, these nodes do not need shared pre-knowledge for authentication, but make use of the exchange of a key-chain for the purpose of authentication. This chain is viewed to be equivalent to the certificate exchange of a public-key scheme, and thereby each value of the chain is used as a Message Authentication Code (MAC) for identification and message authentication. For each node of the communicating parties, it only needs one signature verification and then establishes the trust relationship for each other. Thus, this scheme is not powerful as digital signature, but it indeed reduces the operational complexity.

6.2.2 Centralized Authentication

Centralized authority is always an authentication traditional approach. Wang *et al.* [109] attempt to address the issues occurred by centralized authentication, and so propose a distributed authentication scheme to utilize the trust relations to authenticate neighboring nodes. During the course of authentication, trust transfer is used to authenticate some neighboring nodes through indirect recommendations. When a new node wants to join the network and there are not enough adjacent nodes around the new node, it will randomly choose farther nodes to authenticate it. Such randomness reduces the potential involvement of compromised nodes and is easy to form a reliable certificate chain.

A One-pass Authentication Protocol

Lin *et al.* [67] study the issue of authentication in Universal Mobile Telecommunications System (UMTS), and furthermore, propose a one-pass authentication procedure to realize the IMS (IP multimedia core network subsystem) registration. Their proposed one-pass authentication scheme allows a mobile station (MS) to send a registration request to the serving GPRS support node (SGSN) with a parameter - IP multimedia private identity (*impi*). The call session control function (CSCF) can retrieve another parameter - international mobile subscriber identity (*imsi*) from the SGSN to verify if values of *impi* and *imsi* are consistent, and thereby determine if the authentication is successful or not. Through the analysis of cost, one-pass authentication greatly simplifies the procedure of authentication when compared to two-pass authentication, and save much authentication traffic.

6.2.3 Authentication in Wireless Networks

Qazi *et al.* [90] explore the issue of authentication in a wireless mesh network (WMN) based on AODV routing protocol [86]. In particular, they consider the inclusion of a new user or access point. Those already authenticated servers will generate a ticket with

extra information for the new access point, and then the new access point generates a secret key as the tool of authentication between it and that authenticated server. Savola [95] examines AAA management mechanisms respectively in administrative and self-organized contexts. In a self-organized network, AAA is implemented with Diameter protocol in MANETs, in which each AAA server provides authentication information to its local mobile nodes. Kbar [55] develops a fast token-based authentication technique for WLAN. During the stage of registration, an authentication server will distribute three parameters (a public authentication key, a network access key, and a security token) to those legal mobile clients. The author emphasizes the security of the distribution of these three parameters, since they are only exchanged in the process of registration and kept confidential to the relevant parties, including authentication server, authenticator, and mobile clients.

GLOMONET: a Role Management Authentication Protocol

Hwang and Chang [47] propose a simple authentication technique for the global mobility network (GLOMONET), which attempts to solve the issue of distributed security management by different management roles based on different functions. Totally, two categories of management roles are differentiated: an original manager administrates the long-term authentication key which will be distributed to a user when he makes an authentication request from his home network; while a temporary manager will be responsible for providing roaming authentication services when the user is roaming and asks for authentication from a visited network. Meantime, the temporary manager generates a session key with a short-term timestamp to limit the issued certificate to the roaming user. Figure 6.2 shows the procedure of role assignment with the movement of a user in diverse networks. A user A at home is interacting with its original manager by using a private key PR_A; when this user A moves to an outside network, he communicates with a base station through wireless devices and gets a session key SK_A for authentication.

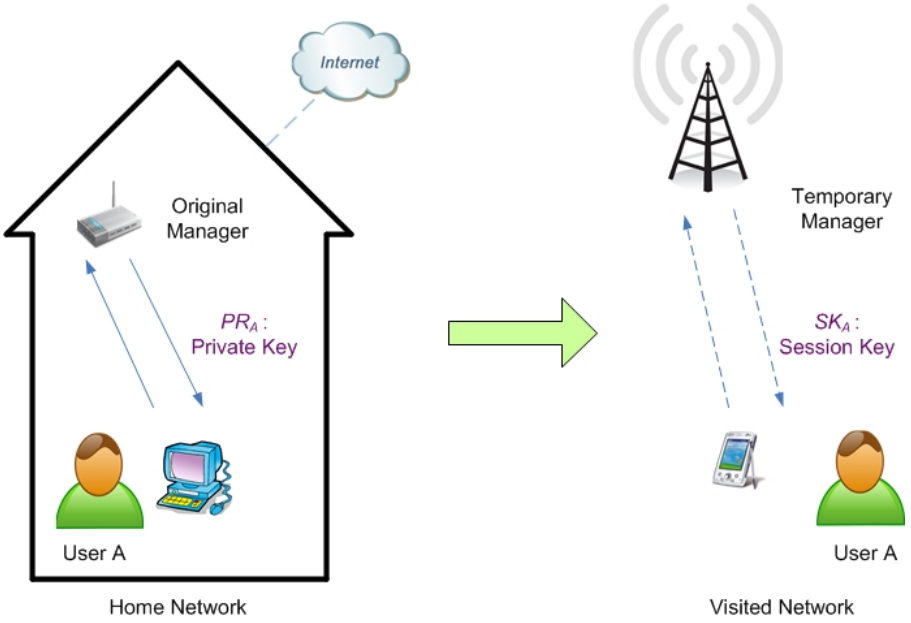


Figure 6.2: The Role Assignment during Different Authentication Processes

6.3 Two-factor Authentication

6.3.1 Authentication Factors

Authentication can identify an entity through some pre-authorized evidences, and thereby grant proper privilege to this authenticated entity. These evidences which can prove an entity's identity are called authentication factors. Traditionally, password is a major authentication method, so a user can register itself to the CA by using the initially assigned password. Nevertheless, password is not the only identification method. With the development of IT, other identification approaches are also available, such as smart card, fingerprint, retina, etc. By means of the analysis of such identification methods, we realize that an authentication factor should have the characteristic of uniqueness within its security domain, in order to verify an entity's identity information. When an authenticator receives such identification information from the user who wants to get

authenticated, it will look up the database or contact to remote central server, to prove the validation of submitted identification information.

Authentication factors can be either physical or virtual. For example, password is a kind of virtual authentication factor; smart card or fingerprint is physical authentication factor. Based on the attributes of each authentication factor, they generally are categorized into "*something you know*", "*something you have*", and "*something you are*". The only requirement for the identifier is that it must be unique when it gets verified. "*Something you know*" can be represented by a password or a personal identification number. "*Something you have*" includes a smart card or security token. "*Something you are*" means the hominine properties of a user, such as fingerprint, retina, or voice characteristics. Through these authentication factors, they provide unique information for an entity to prove his/her identity.

6.3.2 The Concept of Two-factor Authentication

Usually, authentication is to determine whether an entity is actually what it claims. Presently, single-factor and multiple-factor authentications are the mainstreams of authentication. Here, we focus on the method of two-factor authentication.

Two-factor authentication is a widely acknowledged secure authentication strategy which relies on two distinctive factors [31]. As we mentioned aforementioned, one of these two factors is "*something you know*", which can be any identification (e.g. PIN) or virtual token. The other factor is "*something you have*", which could be a certain physically existing evidence to prove the entity identity, such as smart card or biometric access (e.g. fingerprint), as shown in Figure 6.3. By virtue of the combination of logical and physical access, two-factor authentication is able to provide stronger authentication, in comparison with single-factor authentication. In some cases, the factor of "*something you have*" can be replaced by a virtual token to still represent the special sign that the entity possesses.

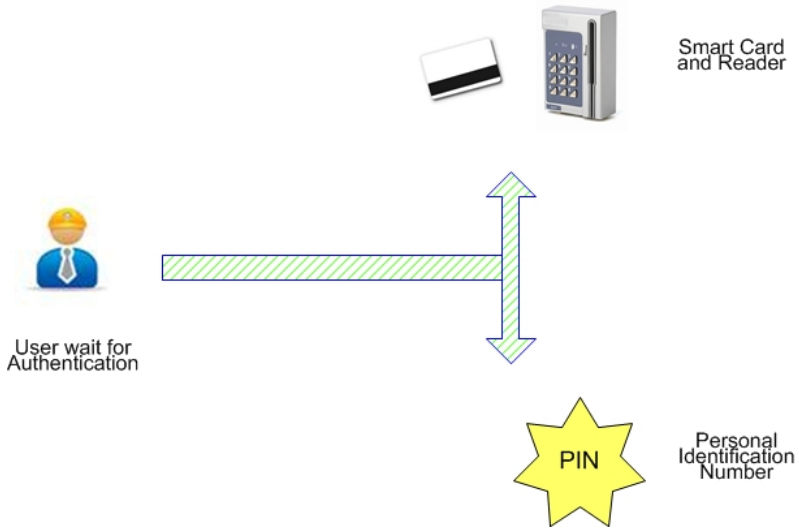


Figure 6.3: A Schematic Diagram of Two-factor Authentication

6.3.3 Motivation

Based on the above discussion of the existing authentication schemes and their applications, we can learn that these solutions have been applied well in the Internet and wired networks. Whereas, we also notice that there are some concerns when applying the traditional authentication methods to wireless ad hoc networks, due to the features of wireless devices. Therefore, before we present our authentication scheme, we would review some issues existing in the current authentication realm and analyze their characteristics, in order to provide an effective applicable solution to a wireless ad hoc network.

Token *vs.* **Physical Factor**

Though the physical factor "*something you have*" is real and it is something we can indeed feel its existence, such as smart card, fingerprint, voice, etc., its deployment is not that easy, in comparison to virtual token. First, the cost of a physical authentication device is more expensive than a virtual token, which is only realized by software. Second, if a smart card is lost or damaged, it needs to re-distribute a new one physically. For a

virtual token, it is easy to replace or revoke an issued token as an authentication factor under such situations.

In a comparison to the virtual token, physical factor is expensive and difficult to manage. As an authentication factor, virtual token is easy to distribute, manage and retrieve. Therefore, virtual token can take over the responsibilities of a physical authentication factor.

The Complex Authentication Procedure

In the existing cryptographic protocols that provide communication security over the Internet, their complexity is a concern when applying them in wireless scenarios, in that such networks need a fast but effective authentication process. SSL and its successor TLS (Transport Layer Security) involve a complicated handshake procedure in client-side and server-side authentication [30]. In SSL and TLS, they encrypt the segments of network connections above the Transport Layer, using asymmetric cryptography for key exchange, symmetric encryption for privacy, and a keyed message authentication code for message reliability. We demonstrate a simple handshake connection example in Figure 6.4.

As shown in the above figure, the server is authenticated by its certificate through the following phases.

(1) Mutual negotiation phase

- client side: a *ClientHello* message which includes TLS version number, a random number, a session ID (*optional*) and cipher methods;
- server side: a response *ServerHello* message, to indicated the agreement about the chosen protocol version, the random number, the resumed session ID (*optional*) and the mutual agreed cipher method;
- server side: issue *Certificate* to the requested client;

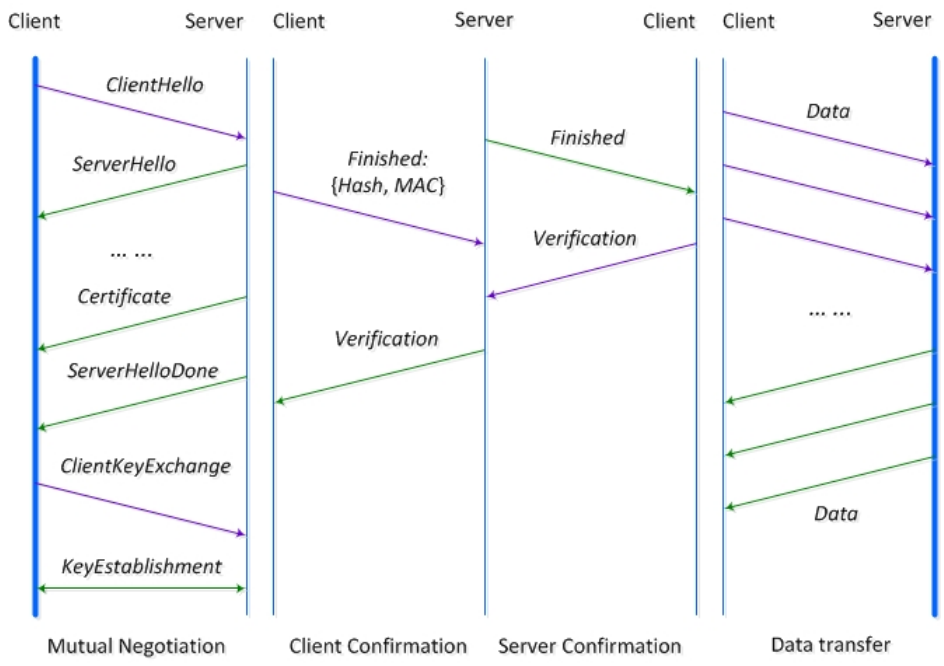


Figure 6.4: A Simple Handshake Procedure with TLS

- server side: a *ServerHelloDone* message which indicates the close of handshake negotiation;
- client side: a *ClientKeyExchange* message to exchange a secret through the public key of the server certificate, and generate the key only shared by the client and server sides.

(2) Client confirmation phase

- client side: an authenticated and encrypted *Finished* message, including a hash and MAC over the previous handshake messages;
- server side: a verification response, attempting to decrypt the *Finished* message, and furthermore verify the provided hash and MAC. If the verification is failed, the connection request is declined.

(3) Server confirmation phase

- server side: it will send its authenticated and encrypted *Finished* message, similar to the client's *Finished* message in the above stage;
- client side: a similar verification response to performs the same decryption and verification operation as the server in the above stage.

(4) Data transfer phase
After the "handshake" is complete and the data transfer phase is initiated, both sides of the application can exchange their official data with encryption and authentication.

This is just a simplified version of TLS authentication between server and client sides. Actually, the real authentication is much more complex than the above illustration. The client and client sides will issue more messages, like *CertificateRequest*, *CertificateVerify*, etc., before they are authenticated each other. Consequently, it is necessary to shorten the procedure of authentication and simplify the complexity of mutual authentication in wireless ad hoc networks.

Wired Equivalent Privacy (WEP)

Another example of authentication is Wired Equivalent Privacy (WEP), an authentication algorithm for IEEE 802.11 wireless networks. WEP primarily employs a shared key authentication method between a WLAN client and an access point (AP). During the authentication process, a challenge-response handshake is built on the following phases:

(1) A client sends an authentication request to its access point;

(2) The AP replies with a plaintext (i.e., cleartext);

(3) The client encrypts the challenge-text using the pre-shared WEP key, and sends it back;

(4) AP decrypts the response and verifies the message. Based on the result of verification, AP will decide if it sends back a positive reply.

It has been pointed out that the challenge messages are a weakness of WEP. Since both sides, client and AP, exchange a plaintext for the authentication, it greatly decreases the security of authentication. This is also another inspiration for the proposal of our token-based authentication scheme.

6.4 A Token-based Two-factor Authentication Scheme

In this section, we focus on the issues of authentication and the application of cryptography in wireless ad hoc networks. When exploring such issues, we propose a token-based two-factor authentication scheme which will authenticate a node twice through two different authentication factors. Additionally, we apply the combination of symmetric and asymmetric key algorithms to form a hybrid cryptosystem. With some preliminaries of community management, we will first explain the theory of how a token is generated. Then, we elaborate how a node will be authenticated with secure credentials and the protection of data transfer in a community.

6.4.1 Token Generation

As we stated in Section 6.3.2, virtual token can replace the authentication factor "*something you have*", because it can simplify the process of physical authentication factor and improve the feasibility of two-factor authentication. In our algorithm, we would employ a polynomial function $f(x)$ to form the virtual tokens. The polynomial function $f(x)$ is represented as follows: $f(x) = c_n x^n + c_{n-1}x^{n-1} + \cdots + c_2 x^2 + c_1 x + a_0$. Here, we denote the constant a_0 as the token T. Thus, $f(x)$ can be expressed as $y = c_n x^n + c_{n-1}x^{n-1} + \cdots + c_2 x^2 + c_1 x + T$, Assume that if the polynomial function $f(x)$ are given sufficient known coefficients $c_n, c_{n-1}, \ldots, c_2, c_1$ and y, it will be able to solve the equation to obtain the token T. Based on such a mechanism, let us indicate one side of authenticated parties which issue a token as *Authentication Subject* (*AS*), and the other side of authenticated parties which will calculate the token as *Authentication Ob-*

ject (AO). For the authentication between AS and AO, AS will generate a unique token T_{AO} for AO. After the AS securely distributes the coefficients c_i or y to AO, AO is able to work out the token T_{AO} through the known c_i or y. Then AO sends the calculated T_{AO} back to AS to get the token verified, so that it can be authenticated. When AS receives the returned T_{AO}, it will check if the received T_{AO} matches with its issued T_{AO}, and thereby determine if AO can be authenticated successfully. Figure 6.5 illustrates such a token-based authentication procedure.

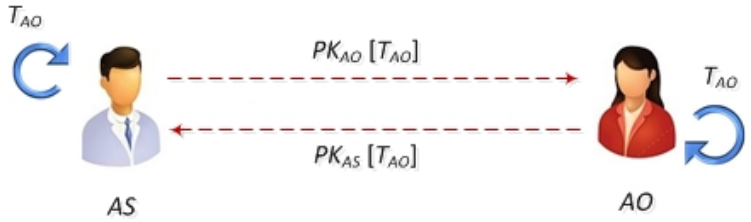


Figure 6.5: The Scheme of Token-based Authentication

6.4.2 A Two-factor Authentication Scheme

For a wireless ad hoc network, as there is not sufficient pre-knowledge about a newly joining node, authentication becomes extremely important to secure the process of membership issue. For any group in our system, its membership is free to any node which behaves well or is not excluded before because of misbehavior. However, the central node will not actively include all of its one-hop neighbors. Hence, in order to join a node's group, the newly joining node has to send a membership request message to the central node to initiate a mutual authentication procedure between them.

First of all, a newly joining node N can easily obtain the public key PK_C of the intended central node C, as PK_C is supposed to be distributed widely. Then it will select a token T which needs to be authenticated by itself. In the meantime, the token T is an integer only known by the node N itself. Since we have explained the course of token generation in detail above, in order to simplify the process of token generation, we

use a quite simple polynomial function to represent the token T. Based on the equation: $y = x^3 + x + T$, N uses its identifier as x to calculate an authentication code y_N, which is encrypted by the public key PK_C of central node C, so that only C can decrypt it.

Subsequently, this node N composes a membership request message which contains not only its identifier ID_N, but also the authentication code y_N, as well as its public key PK_N, for the purpose of their later mutual authentication. Once the central node C receives such a membership request, it will first verify if the identifier is in the list of previous nodes which are excluded due to misbehavior or not. If not, C will use its private key PR_C to decrypt the authentication request y_N, and put the requester's identifier in the equation to calculate the token T.

Finally, after C verified that N did not misbehave before, a secret key (symmetric key) SK_N will be generated by the central node C and encrypted using the public key PK_N of the requester, which is included in the membership request message. Additionally, C will also use the calculated token T and its own identifier ID_C to compute a verification code y_C. Later, the central node C encrypts y_C by using PK_N as well, and composes a membership issue message. This message will contain all information: the encrypted secret key SK_N, the verification code y_C, and its identifier ID_C. C replies it to the newly joining node N, in order to indicate that N has been admitted as a member of the current group. After the node N obtains the response from the central node C, it will use its corresponding private key PR_N to decrypt the secret key SK_N issued from C and save it as its authentication proof with the central node. It will also decrypt the verification code y_C to verify if the token T is correct or not. If yes, it means that the mutual authentication process is successful. Figure 6.6 shows the dual authentication procedure when a newly joining node N attempts to join the group C.

During the process of membership request and issue, in addition to the token, the symmetric and asymmetric keys are combined to act as the tools of authentication as well. First, the pair of public key and private key is the proof of authentication for both communicating sides. Second, the secret key generated by the central node will be unique

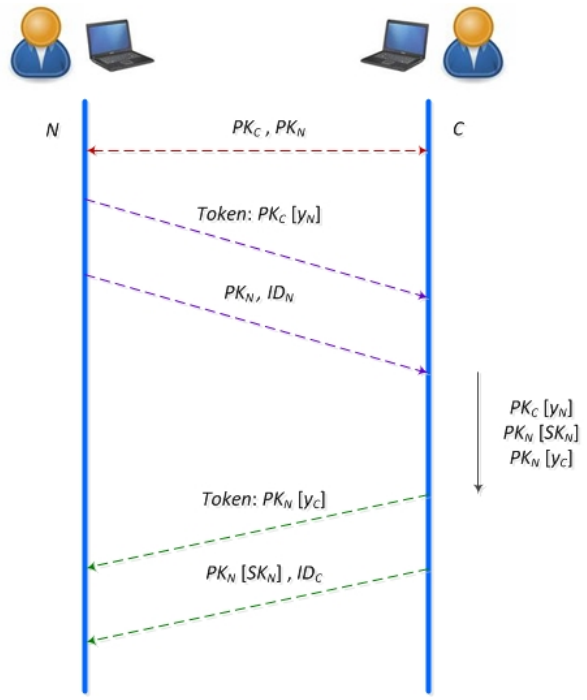


Figure 6.6: The Procedure of Dual Authentication

in its group, and so it is also another proof of authentication. Consequently, indeed this is a multiple factor authentication procedure.

6.4.3 Token and Key Distribution

For the members of its group, the central node will generate a unique secret key for each of its members. Different members will have different secret keys as their own authentication proofs, as well as the cryptographic tool for the communication between them and the central node. Simultaneously, these keys will be updated periodically, so as to be kept as fresh as possible and avoid the exploitation of stale keys. With respect of the update of a secret key, one member's public key is still used to encrypt the updated secret key, and then the encrypted update message is sent to the corresponding member for its key update. Throughout the whole process of data transmission, the public key and unique secret key are combined together to provide effective authentication.

6.4.4 Secure Data Transfer

After a node is accepted as a member of the intended group, it indicates that the newly joined node has been authenticated. Afterwards, any node of the communicating sides (either the central node or members) can initiate the data transfer phase by using the issued unique secret key, and encrypt the data which the sender wants to transfer to the receiver. Once the receiver gets the transferred data, it will decrypt the data using the shared secret key. As we described above, the public key of each member guarantees the secure distribution of each member's unique secret key.

6.5 Security and Performance Analysis

6.5.1 Security Analysis

In this section, we will provide the analysis of security of our two-factor authentication protocol. The methodology and notation related to protocol analysis [40, 82] is utilized here.

Theorem 1. *Public key is able to be widely distributed, so it is available for any node for the use of security protection.*

Proof. As we explained in Section 6.1, public key encryption is a kind of highly security cryptosystem. A pair of keys (PK and PR) consists of such an asymmetric cryptosystem. The public key is exposed and accessible to any party in a network. Based on the asymmetric mechanism, public key PK is not hidden and the distribution of PK does not need to be protected, since the content encrypted by a public key is only decrypted by its corresponding private key PR. The public key can be broadcast at the beginning of network initialization stage or is available on demand. In this way, due to the secrecy of private key, the wide availability of public key will not cause security concerns for data confidentiality.

Lemma 1. *Since a pair of public key and private key is unique to a node, it is capable of being a secret credential for the node.*

Proof. In PKI, the public key and private key are unique for a node in a specific domain. That is, in a network, once a pair of public key and private is assigned to a certain node, the same public key and private key will not be assigned to other nodes. According to the requirement as an authentication factor, uniqueness is extremely important for this pair of keys, so that they can become a type of secret credential for that node. At the same time, due to the coupling of public key and private key, they provide reliable security for data protection. In most of the existing related works, PKI is always supposed to be secure about their ability for data confidentiality. For example, if a node X would establish an authenticated channel with another node Y, the node X will send its public key PK_X to Y, and thereby, Y replies a credential issue message protected by PK_X to X. During the process, if a malicious node M intercepts or impersonates X's information, M will not decipher the replied credential by using PK_X, because it does not know the corresponding private key PR_X. Here, PR_X is apparently a unique and confidential credential for node X. Consequently, due to the uniqueness of public key and private key, they can be viewed as a credential for a node in the network.

Lemma 2. *In two-factor authentication, a token will be unique for each node.*

Proof. In our scheme, we employ a virtual token to act as the other authentication factor in two-factor authentication. When generating a token, the authenticator will keep the token T as unique in its domain, so that each node will have a different token as its own proof to the authenticator. For instance, when a node Y generates a token T_X for another node X, it will distribute this token to X in a secure way as stated below, so that malicious node M cannot access the T_X at all. Thus, our virtual token mechanism meets the requirement of uniqueness as an authentication factor.

Theorem 2. *These two factors satisfy the requirement of uniqueness for an authentication system, and can achieve the authentication between a pair of nodes.*

Proof. Based on the above discussion, we can learn that the pair of public key and

private key is unique for its owner, and at the same time, the token generated by the authenticator is unique as well to one node. Both of these factors possess the characteristics of uniqueness and satisfy the requirement of authentication. Additionally, public key ensures the confidentiality of exchanged information. Malicious nodes cannot obtain the authenticated information or credentials. Therefore, such a two-factor authentication scheme is able to achieve the authentication between a pair of nodes.

Theorem 3. *The token is not released directly, and it can not be obtained without pre-knowledge.*

Proof. In order to enhance the security of the virtual token used in our two-factor scheme, the authenticator does not directly release the token to its destined node. Through applying a polynomial function, the token is hidden in the result of the computation of polynomial function. When the destined node receives such a token with cryptographic protection, it can calculate the token from the polynomial function based on their pre-distributed knowledge about the function. Thus, even if some malicious nodes intercept the token distribution, they still have no idea of the token directly, assumed that they cannot learn the pre-distribution knowledge about the function. Hence, it provides the protection for the authentication factor in an implicit way.

Theorem 4. *The use of public key guarantees the process of token distribution is secure.*

Proof. During the procedure of token distribution, the employment of the public key of the destined node can provide protection against the attacks for token distribution. This is why a newly joined node X needs to send its public key PK_X first when it moves into the neighborhood of node Y. Since the node Y in a community maintains the public keys PK of all members, it will use the corresponding public key of each neighboring member X to encrypt the token T_X generated in the authentication phase. Subsequently, the encrypted token $PK_X[T_X]$ will be sent along a unicast route to each corresponding member and only the destined node X will be able to decrypt the distributed token message $PK_X[T_X]$ using its own private key PR_X. As a result, the process of token

distribution is guaranteed to be secure and avoids the attempt of malicious nodes to decipher the distributed token.

Theorem 5. *The exchange of public key provides mutual authentication for the communicating parties.*

Proof. During the registration and admission procedures between a pair of nodes, the authenticated parties will exchange their public keys in the process of communication. First, the initiating node I will send its public key PK_I to the node H to require the authentication from H. Subsequently, if the node H admits node I and then it will let the node I know its public key PK_H as well. Here, a node's public key plays an important role not only in the protection of exchanged data, but also in the mission of acting as an authentication factor. They both benefit from the advantage of the availability in PKI. Obviously, it is a mutual authentication process for both sides of the communicating parties.

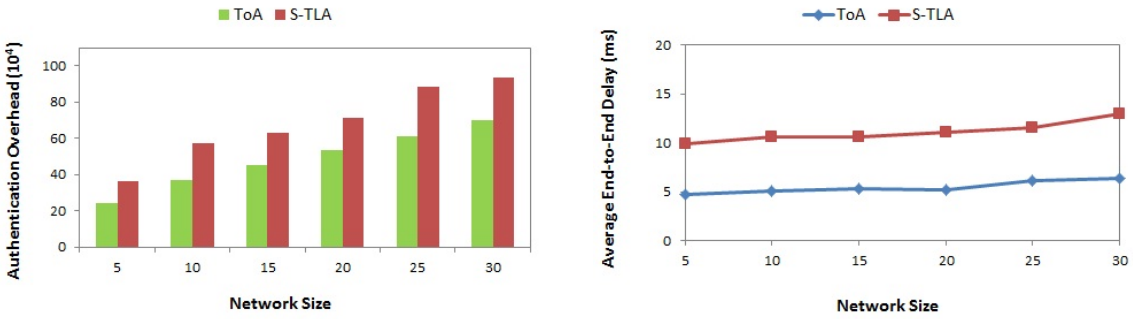
Theorem 6. *The randomness of token generation increases the flexibility of authentication factor.*

Proof. One of the advantages of virtual token is that a token is randomly generated. Unlike the authentication factors "*something you have*" or "*something you are*", they cannot be changed or are inconvenient to change, for instance, fingerprint, smart card, etc. However, virtual token is determined just before the start of authentication stage. That is, these tokens are not generated until the authenticator receives such a request. Additionally, the authentication factor "token" has the characteristics of randomness and especially is verifiable by means of a certain calculation and pre-knowledge. Thus, our proposed virtual token mechanism is more flexible and greatly enhances the reliability of authentication.

6.5.2 Performance Analysis

In order to study the characteristics of our proposed scheme and know how it performs within a wireless ad hoc environment in *ns2* [49]. We evaluate the proposed token-based

authentication scheme through the simulations, whose experimental settings are similar to Section 5.5.2. We have chosen the following performance metrics for evaluating our system. 1) *Average End-to-End Delay*: records the average time spent on the process of authentication. 2) *Average Authentication Overhead*: means the average amount of packets used for authentication, 3) *Authentication Percentage*: is the percent of packets sent for authentication, when compared to the total amount of packets. 4) *Ratio of Malicious Nodes*: indicates the proportion of malicious nodes exist in a network. 5) *Average Amount of Community Members*: represents the average number of neighbors in the community within a network.

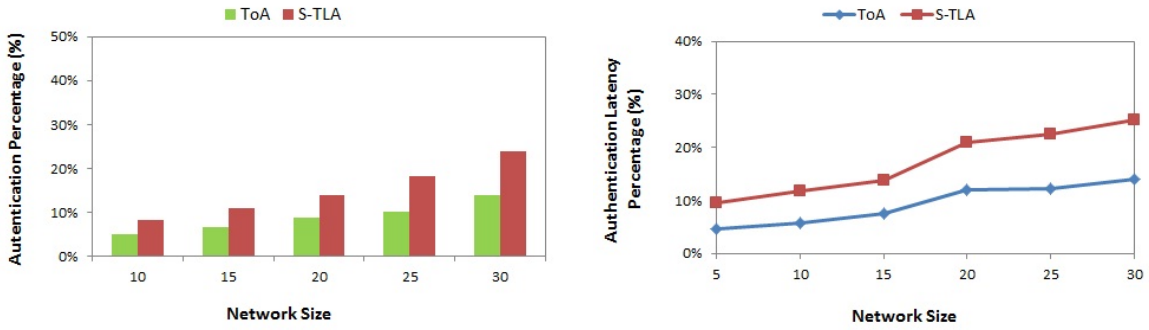


(a) (b)

Figure 6.7: End-to-End Delay

In contrast to our scheme, we present an authentication system which is a basic version of TLS as we described in Section 6.3.3. Here, we name this comparable TLS authentication scheme as S-TLA. In the Figure 6.7, we compare these two authentication schemes to measure their overhead and latency respectively spent on authentication. Figure 6.7 (a) examines the total overhead of authentication for both schemes. When the size of network increases, the total overhead of ToA and S-TLA spent on authentication increase as well. We can note that ToA has a slower increase than S-TLA since it uses a simplier authentication process and has less handshake communication. Figure 6.7 (b) shows the end to end delay between a central node and its members for their authentication. Though the network size increases, the average time spent on our ToA scheme

is less than S-TLA due to the same reason. That is, ToA does not initiate handshake communication more often than S-TLA and less time is spent on the process of authentication and handshake. From the above figure, we can see that a simplified authentication procedure can reduce the cost and latency of authentication.

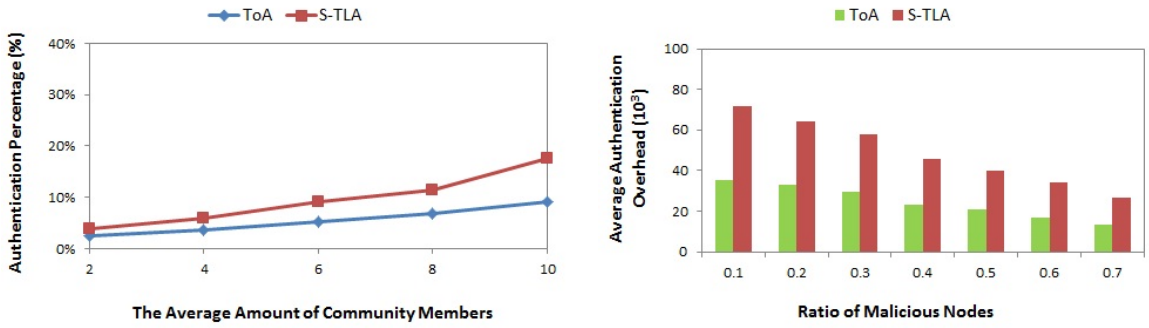


(a) (b)

Figure 6.8: Average Authentication Percentage

Subsequently, in Figure 6.8 (a), the average authentication overhead is examined. We can notice that the authentication percentage increases slightly with the increase of network size and does not have a heavier weight on the total overhead. As such, Figure 6.8 (b) investigates the comparison of latency percentage between ToA and S-TLA. It is obvious that ToA owns a less latency percentage that S-TLA, in that it has less handshake times than S-TLA. Consequently, our authentication scheme is indeed a light-weighted authentication solution for a wireless and ad hoc network, and thereby, it is suitable for such an environment.

Additionally, we investigate the changes of overhead spent on the authentication with the increase of community members. In the Figure 6.9 (a), we can see that the overhead spent on authentication will increase with the increase of average community members. which means that a central node needs spend more overhead on authentication, if it has more members in its community. Thus, the amount of a community's members can have a direct influence on the authentication overhead. Figure 6.9 (b) shows the effect from the ratio of malicious nodes. The average overhead of authentication decreases with the



(a) (b)

Figure 6.9: The Factors that Influence the Overhead of Authentication

increase of the proportion of malicious nodes, whichever authentication strategy is chosen, either ToA or S-TLA. This is caused by the reason that malicious nodes will not actively respond to the authenticatin requests from the central nodes, so less packets are involved with the more inclusion of malicious nodes. Through the analysis about the factors that affect the overhead of authentication, we learn that what kind of authentication mechanism is applied has an important influence on the performance of a network.

6.6 Summary

Authentication is a significant component in securing a network. In this chapter, we study the application of two-factor authentication, and thereby, we present a solution based on two-factor authentication. First, our scheme takes advantage of a node's public key as an authentication factor; on the other hand, a virtual token is adopted to act as the other factor of two-factor authentication. Thus, when an authenticator validates the identification of a node, it can verify the node through two different factors, so that the reliability of authentication is improved. Through our analysis, our scheme offers flexible protection to dynamic and agile networks, and demonstrates that two-factor authentication is able to provide satisfactory authentication for secure wireless communication.

Chapter 7

Conclusion and Future Works

7.1 Final Remarks

In recent years, security has been one of the key challenges in wireless ad hoc networks. Especially, wireless networks have their own features and requirements for security. This thesis has presented several computational models for the design of secure and trustworthy networks which are able to make rational decisions when encountering potential threats. To this end, we harmonized new and emerging technologies with these special security requirements.

Group management demonstrates a promising paradigm to mitigate the effect of bad-behaved members and to prevent the occurrence of malicious behavior. In particular, secure group management ensures the functions of the entire networks and offers reliable security protection. A secure and distributed network management model for wireless networks involves multiple security techniques, from trust to cryptography, to authentication. Based on our community model, many important security issues are simplified; this model furthermore reduces the complexity of traditional group management methods.

The concept of trust provides the foundation of decentralized management and makes misbehavior detection feasible. Traditional trust models do not actively encourage cooperation among nodes in wireless ad hoc networks, which is particularly significant for

such networks. A novel trust model that can compute the trust value of a node based on the node's past behavior, greatly changes the current computational methodology and ideology of trust. Thus, the nodes throughout the entire network are managed in a distributed, dynamic, and effective manner.

Traditional encryption algorithms are employed to protect data confidentiality and integrity. However, due to the strengths and weaknesses of these algorithms, the application of these algorithms to wireless ad hoc networks is not well determined. An advisable solution for the application of these cryptographic algorithms is proposed, together with a selective encryption scheme. Here, we attempt to use a combination of symmetric and asymmetric cryptography to achieve secure data communication. At the same time, this proposed selective encryption algorithm can reduce the computational costs for wireless devices. In addition, based on the concept of two-factor authentication, we take advantage of a virtual token to establish a two-factor authentication, so that the reliability of authentication is improved.

Through the security analysis and experimental results, we have shown how decentralized management is useful in wireless and ad hoc scenarios, and how trust provides feasible solutions for misbehavior detection. We believe that wireless ad hoc applications or other dynamic and agile systems can benefit from our secure and distributed models.

7.2 The Contributions of this Thesis

We explored several important security issues in wireless ad hoc networks, and a summary of the main contributions of this thesis is listed below.

- We introduced the concept of community and propose the SeDi community-based model. This model allows the involvement of other techniques, such as trust, authentication, cryptography, etc.; and it can protect the security of each local group in a network. Moreover, our SeDi model not only avoids the use of centralized authority, but also manages the network in a fully distributed manner.

- A trust-based security system, which is referred to as TOMS, is proposed and examined particularly with respect to the security of wireless ad hoc settings. TOMS defines the concept of trust, establishes the trust relationship between distributed nodes, involves the novel and effective computational model, and specifies a set of rules in this system for wireless nodes. TOMS totally changes the traditional linear-based trust computation method. In so far as we are aware, this is one of the very first approaches that calculates the trust value based on different increase-shapes.

- The design of a prediction mechanism for the reliability of indirect trust recommendation and a node evaluation scheme with assistant trust (NEAT), aim to mitigate the effect caused by some specific misbehavior as we described in Chapter 4. On the other hand, a supervised learning scheme based on Naïve Bayes is designed to judge the reliability of information provided in relation to trust. To our best knowledge, Naïve Bayes is rarely applied for the detection of misbehavior along with trust, and this is one of the few attempts that combine Naïve Bayes with trust for the prediction of reliability in wireless ad hoc networks.

- We investigated the existing symmetric and asymmetric encryption algorithms and provided an advisable solution (HiC) for the application of traditional cryptographic algorithms. This solution applies each type of encryption algorithm in the right scenarios according to its own strengths; and a reasonable trade-off between these algorithms is found to balance the security and computational costs.

- A selective encryption algorithm is designed to economize the overhead spent on data protection and to prevent data disclosure to untrustworthy nodes. This algorithm encrypts the transmitted packets by means of probabilistic function and involves the uncertainty to data encryption. Thereby, it can enhance the reliability of selective algorithms.

- We developed a token-based authentication mechanism (ToA) based on the concept of two-factor authentication. Our scheme takes advantage of a node's public key

as an authentication factor; on the other hand, a virtual token is employed to act as the other factor in a two-factor authentication process. Thus, ToA eliminates the weakness of single-factor authentication, and thus enhances the reliability of authentication.

7.3 Future Research Directions

In this thesis work, we have realized several possible future research directions, and they may enhance the proposed solutions:

- We will investigate possibilities for the integration of different security strategies to support the future generation of the wireless network management model that intends to explore fully the replacement of centralized management, deal more with malicious behavior, and enhance the security of a distributed network. This integration will not only protect data communication among trustworthy users, but it will also provide extendable security support for new applications.

- The techniques of data mining and machine learning are widely applied in lots of realms, because they can provide reliable help in the process of decision making and data processing. There are several directions that could be considered to facilitate the process of intrusion detection and prevention. We plan to investigate some aspects for our proposed misbehavior detection schemes based on the applications of data mining and machine learning.

- The issue of digital certificate is important for wireless ad hoc networks. We plan to explore the potential applications and inclusions for our proposed strategies and protocols.

- Although we studied the performance of our mechanisms through simulation experiments, we will consider implementing our proposed security strategies in a testbed for the assessment of their performance in real scenarios.

Appendix A

List of Publications Related to Thesis

Submitted Conference/Journal Papers

- Yonglin Ren, Azzedine Boukerche, "A Secure and Efficient Data Transmission Mechanism based on Authentication for Wireless E-healthcare Systems", submitted to *IEEE Transactions on Information Technology in Biomedicine*, 2011.

Journal Papers (Published)

- Yonglin Ren, Richard Werner Nelem Pazzi, and Azzedine Boukerche, "Monitoring Patients via a Secure and Mobile Healthcare System", *IEEE Wireless Communications*, Vol. 17(1), pp. 59-65, 2010.
- Yonglin Ren, Azzedine Boukerche, "ARMA: a Scalable Secure Routing Protocol with Privacy Protection for Mobile Ad Hoc Networks", *Wireless Communications and Mobile Computing*, Wiley & Sons, Vol. 10(5), pp. 672-687, 2010.
- Azzedine Boukerche, Yonglin Ren, "A Secure Mobile Healthcare System using Trust-Based Multicast Scheme", *IEEE Journal on Selected Areas in Communications*, Vol. 27(4), pp. 387-399, 2009.
- Azzedine Boukerche, Yonglin Ren, "A Trust-Based Security System for Ubiquitous and Pervasive Computing Environments", *Computer Communications*, Elsevier, Vol. 31(18), pp. 4343-4351, 2008.

Conference Papers (Published)

- Yonglin Ren, Azzedine Boukerche, and Lynda Mokdad, "Performance Analysis of a Selective Encryption Algorithm for Wireless Ad Hoc Networks", *Proceedings of IEEE Wireless Communications and Networking Conference* (WCNC), pp. 1038-1043, March 28-31, 2011.

- Yonglin Ren, Azzedine Boukerche, and Richard Werner Nelem Pazzi, "Performance Evaluation of a Hybrid Cryptosystem with Authentication for Wireless Ad hoc Networks", *Proceedings of the IEEE Global Communications Conference* (GLOBECOM), pp. 1-5, December 6-10, 2010.

- Yonglin Ren, Azzedine Boukerche, "A Secure Group Management Scheme for Mobile Ad Hoc Networks", *Proceedings of the 15th IEEE Symposium on Computers and Communications* (ISCC), pp. 429-432, June 22-25, 2010.

- Yonglin Ren, Azzedine Boukerche, "Performance Analysis of Trust-Based Node Evaluation Schemes in Wireless and Mobile Ad Hoc Networks", *Proceedings of IEEE International Conference on Communications* (ICC), pp. 1-5, June 14-18, 2009.

- Yonglin Ren, Richard Werner Nelem Pazzi, and Azzedine Boukerche, "Weighted-NEAT: An Efficient Weighted Node Evaluation Scheme with Assistant Trust Mechanisms to Secure Wireless Ad Hoc Networks", *Proceedings of 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks* (WOWMOM), pp. 1-8, June 15-19, 2009.

- Yonglin Ren, Azzedine Boukerche, "An Efficient Trust-Based Reputation Protocol for Wireless and Mobile Ad Hoc Networks: Proof and Correctness", *Proceedings of the IEEE Global Communications Conference* (GLOBECOM), pp. 1892-1896, November 30 - December 4, 2008.

- Yonglin Ren, Azzedine Boukerche, "Modeling and Managing the Trust for Wireless and Mobile Ad Hoc Networks", *Proceedings of IEEE International Conference on Communications* (ICC), pp. 2129-2133, May 19-23, 2008.

- Azzedine Boukerche, Yonglin Ren, "The Design of a Secure Key Management System for Mobile Ad Hoc Networks", *Proceedings of the 33rd IEEE Conference on Local Computer Networks* (LCN), pp. 320-327, October 14-17, 2008.

- Azzedine Boukerche, Yonglin Ren, "A Security Management Scheme using a Novel Computational Reputation Model for Wireless and Mobile Ad Hoc Networks", *Proceedings of the 5th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks* (PE-WASUN), pp. 88-95, October 27-28, 2008.

Bibliography

[1] W. Aiello, S. M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A. D. Keromytis, and O. Reingold. Just fast keying: Key agreement in a hostile internet. *ACM Transactions on Information and System Security*, Vol. 7:242–273, 2004.

[2] M. Aikawa, K. Takaragi, S. Furuya, and M. Sasamoto. A lightweight encryption method suitable for copyright protection. *IEEE Transactions on Consumer Electronics*, Vol. 44:902–910, 1998.

[3] J. N. Al-Karaki and A. E. Kamal. Stimulating node cooperation in mobile ad hoc networks. *Springer Wireless Personal Communications*, Vol. 44:219–239, 2007.

[4] A.-R. Alfarez and S. Hailes. A distributed trust model. In *Proceedings of the ACM workshop on New security paradigms*, pages 48–60, 1998.

[5] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh. Max-min d-cluster formation in wireless ad hoc network. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 32–41, 2000.

[6] G. Autran and X. Li. Large scale deployment a mobile agent approach to network management. In *Proceedings of 7th IEEE International Conference on Networking*, pages 614–619, 2008.

[7] V. Balakrishnan, V. Varadharajan, U. K. Tupakula, and P. Lucs. Trust and recommendations in mobile ad hoc networks. In *Proceedings of the Third IEEE International Conference on Networking and Services*, 2007.

[8] S. Bandyopadhyay and E. J. Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1713–1723, 2003.

[9] F. Bao and R. H. Deng. Light-weight encryption schemes for multimedia data and high-speed networks. In *Proceedings of IEEE Global Telecommunications Conference*, pages 271–350, 2007.

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

[11] M. Bohge and W. Trappe. An authentication framework for hierarchical ad hoc sensor networks. In *Proceedings of the 2nd ACM workshop on Wireless security*, pages 79–87, 2003.

[12] L. Bononi and C. Tacconi. A wireless intrusion detection system for secure clustering and routing in ad hoc networks. In *Proceedings of 9th Springer International Conference on Information Security*, pages 398–414, 2006.

[13] D. Bottazzi, A. Corradi, and R. Montanarii. A context-aware group management middleware to support resource sharing in manet environments. In *Proceedings of 6th ACM international conference on Mobile data management*, pages 147–151, 2005.

[14] A. Boukerche. *Handbook of Algorithms for Wireless and Mobil Networks and Computing*. CRC Chapman Hall, 2005.

[15] A. Boukerche. *Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks*. Wiley and Sons, 2008.

[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. *Elsevier Computer Communications*, Vol. 28:1193–1203, 2005.

[17] A. Boukerche and Y. Ren. Arma: An efficient secure ad hoc routing protocol. In *Proceedings of the IEEE Global Communications Conference*, pages 1268–1272, 2007.

[18] 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 *Proceedings of the ACM/IEEE Annual International Conference on Mobile Computing and Networking*, pages 85–97, 1998.

[19] S. Buchegger. *Coping with Misbehavior in Mobile Ad-hoc Networks*. PhD thesis, EPFL, CRC Press, 2004.

[20] S. Buffett, M. W. Fleming, M. M. Richter, N. Scott, and B. Spencer. Determining internet users' values for private information. In *Proceedings of Second Annual Conference on Privacy, Security and Trust*, pages 79–88, 2004.

[21] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. In *Proceedings of the 12th ACM Symposium on Operating Systems Principles*, pages 18–36, 1989.

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

[23] A. C.-F. Chan. Distributed symmetric key management for mobile ad hoc networks. In *Proceedings of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 2414–2424, 2004.

[24] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 197–213, 2003.

[25] K. Chang and K. G. Shin. Distributed authentication of program integrity verification in wireless sensor networks. *ACM Transactions on Information and System Security*, Vol. 11, 2008.

[26] C. Chen, J. Zhang, R. Cohen, and P.-H. Ho. Secure and efficient trust opinion aggregation for vehicular ad-hoc networks. In *Proceedings of 72nd IEEE Vehicular Technology Conference*, pages 1–5, 2010.

[27] H. Chen, H. Wu, X. Cao, and C. Gao. Trust propagation and aggregation in wireless sensor networks. In *Proceedings of Japan-China Joint IEEE Workshop on Frontier of Computer Science and Technology*, pages 13–20, 2007.

[28] K. Cheng, J. Luo, and C. Zhang. Rough set weighted nave bayesian classifier in intrusion prevention system. In *Proceedings of IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing*, pages 25–28, 2009.

[29] Wikipedia contributors. Naive bayes classifier. Wikipedia, The Free Encyclopedia, Last accessed in Jan. 2011. http://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[30] Wikipedia contributors. Transport layer security. Wikipedia, The Free Encyclopedia, Last accessed in Mar. 2010. http://en.wikipedia.org/wiki/Transport_Layer_Security.

[31] Wikipedia contributors. Two-factor authentication. Wikipedia, The Free Encyclopedia, Last accessed in Mar. 2010. http://en.wikipedia.org/wiki/Two-factor_authentication.

[32] H. Dahshan and J. Irvine. On demand self-organized public key management for mobile ad hoc networks. In *Proceedings of IEEE 69th Vehicular Technology Conference*, pages 1–5, 2009.

[33] D. Damodaran, R. Singh, and P. D. Le. Group key management in wireless networks using session keys. In *Proceedings of 3rd IEEE International Conference on Information Technology: New Generations*, pages 402–407, 2006.

[34] H. David, M. Heikki, and S. Padhraic. *Principles of Data Mining*. MIT Press, Cambridge, 2001.

[35] Q. Ding, X. Li, M. Jiang, and X. Zhou. Reputation-based trust model in vehicular ad hoc networks. In *Proceedings of IEEE International Conference on Wireless Communications and Signal Processing*, pages 1–6, 2010.

[36] N. Dragoni, F. Massacci, C. Schaefer, T. Walter, and E. Vetillard. A security-by-contract architecture for pervasive services. In *Proceedings of the 3rd IEEE International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, pages 49–54, 2007.

[37] C.-I Fan and Y.-H. Lin. Provably secure remote truly three-factor authentication scheme with privacy protection on biometrics. *IEEE Transactions on Information Forensics and Security*, Vol. 4:933–945, 2009.

[38] S. Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly and Associates, 1995.

[39] T. Ghosh, N. Pissinou, and K. Makki. A framework for computing trust in mobile ad-hoc networks. *Springer Mobile and Wireless Network Security and Privacy*, pages 67–83, 2007.

[40] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pages 234–248, 1990.

[41] F. J. Groen, C. S. Smidts, A. Mosleh, and S. Swaminathan. Qras - the quantitative risk assessment system. In *Proceedings of Annual IEEE Reliability and Maintainability Symposium*, pages 349–355, 2002.

[42] L. Hao, X. Li, and S. Yang. Fast authentication public key infrastructure for mobile ad hoc networks based on trusted computing. In *Proceedings of IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, pages 1–4, 2006.

[43] F. M. Heikkila. Encryption: Security considerations for portable media devices thin films and exchange anisotropy. *IEEE Security and Privacy*, Vol. 5:22–27, 2007.

[44] H. Hofbauer and A. Uhl. Selective encryption of the mc ezbc bitstream for drm scenarios. In *Proceedings of the 11th ACM workshop on Multimedia and security*, pages 161–170, 2009.

[45] D. L. Hoffman, T. P. Novak, and M. Peralta. Building consumer trust online. *Communications of the ACM*, Vol. 42:80–85, 1999.

[46] D. Huang and D. Medhi. A key-chain-based keying scheme for many-to-many secure group communication. *ACM Transactions on Information and System Security*, Vol. 7:523–552, 2004.

[47] K.-F. Hwang and C.-C. Chang. A self-encryption mechanism for authentication of roaming and teleconference services. *IEEE Transactions on Wireless Communications*, Vol. 2:400–407, 2003.

[48] O. Ileri, S.-C. Mau, and N. B. Mandayam. Pricing for enabling forwarding in self-configuring ad hoc networks. *IEEE Journal on Selected Areas in Communications*, Vol. 23:151–162, 2005.

[49] University of Southern California Information Sciences Institute. Ns-2, Last accessed in May 2011. http://www.isi.edu/nsnam/ns/.

[50] T. Jamil. The rijndael algorithm. *IEEE Potentials*, Vol. 23:36–38, 2004.

[51] L. Jun, L. Zou, C. Xie, and H. Huang. A two-way selective encryption algorithm for mpeg video. In *Proceedings of IEEE International Workshop on Networking, Architecture, and Storages*, 2006.

[52] L. Kagal, T. Finin, and A. Joshi. Trust-based security in pervasive computing environments. *IEEE Computer*, Vol. 34:154–157, 2001.

[53] K. Kane and J. C. Browne. Using uncertainty in reputation methods to enforce cooperation in ad-hoc networks. In *Proceedings of the 5th ACM workshop on Wireless security*, pages 105–113, 2006.

[54] T. Kaya, G. Lin, G. Noubir, and A. Yilmaz. Secure multicast groups on ad hoc networks. In *Proceedings of 1st ACM workshop on Security of ad hoc and sensor networks*, pages 94–102, 2003.

[55] G. Kbar. Wireless network token-based fast authentication. In *Proceedings of 17th IEEE International Conference on Telecommunications*, pages 227–233, 2010.

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

[57] R. Klempous, J. Nikodem, L. Radosz, and N. Raus. Adaptive misbehavior detection in wireless sensors network based on local community agreement. In *Proceedings of 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, pages 153–160, 2007.

[58] M. Kochhal, L. Schwiebert, and S. Gupta. Role-based hierarchical self organization for wireless ad hoc sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 98–107, 2003.

[59] R. Ksters and M. Tuengerthal. Computational soundness for key exchange protocols with symmetric encryption. In *Proceedings of 16th ACM conference on Computer and communications security*, pages 91–100, 2009.

[60] C. Labonte and S. Srinivas. Group management strategies for secure multicasting on activevirtual private networks. In *Proceedings of 25th IEEE Conference on Local Computer Networks*, pages 213–222, 2000.

[61] K. Lee. A group communication protocol architecture for distributed network management systems. In *Proceedings 4th IEEE International Conference on Computer Communications and Networks*, pages 28–31, 1995.

[62] W. Li and A. Joshi. Outlier detection in ad hoc networks using dempster-shafer theory. In *Proceedings of the 10th IEEE International Conference on Mobile Data Management*, pages 112–121, 2009.

[63] W. Li, J. Parker, and A. Joshi. Security through collaboration in manets. In *Proceedings of 4th Springer's International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 696–714, 2008.

[64] S. Lian, Z. Liu, Z. Ren, and H. Wang. Secure advanced video coding based on selective encryption algorithms. *IEEE Transactions on Consumer Electronics*, Vol. 52:621–629, 2006.

[65] H.-C. Liao, P.-C. Lee, Y.-H. Chao, and C.-L. Chen. A location-dependent data encryption approach for enhancing mobile information system security. In *Proceedings of the 9th IEEE International Conference on Advanced Communication Technology*, pages 625–628, 2007.

[66] I-E. Liao, C.-C. Lee, and M.-S. Hwang. A password authentication scheme over insecure networks. *Elsevier Journal of Computer and System Sciences*, Vol. 72:727–740, 2006.

[67] Y.-B. Lin, M.-F. Chang, M.-T. Hsu, and L.-Y. Wu. One-pass gprs and ims authentication procedure for umts. *IEEE Journal on Selected Areas in Communications*, Vol. 23:1233–1239, 2005.

[68] H. Liu and B. Krishnamachari. A price-based reliable routing game in wireless networks. In *Proceedings of the ACM workshop on Game theory for communications and networks*, 2006.

[69] J. Liu, F. Sailhan, D. Sacchetti, and V. Issarny. Group management for mobile ad hoc networks: Design, implementation and experiment. In *Proceedings of the ACM 6th international conference on Mobile data management*, pages 192–199, 2005.

[70] K. Liu and J. Li. Mobile cluster protocol in wireless ad hoc networks. In *Proceeding of IEEE International Conference on Communication Technology*, pages 568–573, 2000.

[71] L. Luo, R. Safavi-Naini, J. Baek, and W. Susilo. Self-organised group key management for ad hoc networks. In *Proceedings of the ACM Symposium on Information, computer and communications security*, pages 138–147, 2006.

[72] P. Maggi and R. Sisto. A configurable mobile agent data protection protocol. In *Proceedings of the 2nd ACM international joint conference on Autonomous agents and multiagent systems*, pages 851–858, 2003.

[73] P. Marbach and Y. Qiu. Cooperation in wireless ad hoc networks: a market-based approach. *IEEE/ACM Transactions on Networking*, Vol. 13:1325–1338, 2005.

[74] S. Marti, T. J. Giuli, K. Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of 6th ACM annual international conference on Mobile computing and networking*, pages 255–265, 2000.

[75] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, and F.-O. Devaux. Secure and low cost selective encryption for jpeg2000. In *Proceedings of 10th IEEE International Symposium on Multimedia*, pages 31–38, 2008.

[76] J. McHugh and J. B. Michael. Secure group management in large distributed systems: What is a group and what does it do? In *Proceedings of the ACM workshop on New security paradigms*, pages 80–85, 1999.

[77] D. H. McKnight and N. L. Chervany. The meanings of trust. Technical report, University of Minnesota, 1996.

[78] A. Mondal and M. Kitsuregawa. Privacy, security and trust in p2p environments: A perspective. In *Proceedings of 17th IEEE International Conference on Database and Expert Systems Applications*, pages 682–686, 2006.

[79] G. Montenegro and C. Castelluccia. Crypto-based identifiers (cbids): Concept and applications. *ACM Transaction on Information and System Security*, Vol. 7:97–127, 2004.

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

[81] E. C. H. Ngai and M. R. Lyu. An authentication service based on trust and clustering in wireless ad hoc networks: Description and security evaluation. In *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, pages 94–103, 2006.

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

[83] S. Park, B. Aslam, and C. C. Zou. Long-term reputation system for vehicular networking based on vehicle's daily commute routine. In *Proceedings of IEEE Consumer Communications and Networking Conference*, pages 436–441, 2011.

[84] E. Pat-Cornell. Finding and fixing systems weaknesses: Probabilistic methods and applications of engineering risk analysis. *Wiley's Risk Analysis*, Vol. 22:319–334, 2002.

[85] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang. A wireless hierarchical routing protocol with group mobility. In *Proceedings of IEEE Wireless Communications and Networking Conference*, pages 1538–1542, 1999.

[86] C. E. Perkins and M. E. Royer. Ad hoc on demand distance vector (aodv) routing. IETF Internet Draft, 1997. www.ietf.org.

[87] A. A. Pirzada and C. McDonald. Kerberos assisted authentication in mobile ad-hoc networks. In *Proceedings of the 27th ACM Australasian conference on Computer science*, pages 41–46, 2004.

[88] U. Potdar, K. T. Talele, and S. T. Gandhe. Comparison of mpeg video encryption algorithms. In *Proceedings of ACM International Conference on Advances in Computing, Communication and Control*, pages 289–294, 2009.

[89] K. Prakobphol and J. Zhan. A novel outlier detection scheme for network intrusion detection systems. In *Proceedings of the IEEE International Conference on Information Security and Assurance*, 2008.

[90] S. Qazi, Y. Mu, and W. Susilo. Securing wireless mesh networks with ticket-based authentication. In *Proceedings of 2nd IEEE International Conference on Signal Processing and Communication Systems*, pages 1–10, 2008.

[91] R. K. Rainer, C. A. Snyder, and H. H. Carr. Risk analysis for information technology. *Journal of Management Information Systems*, Vol. 8:129–147, 1991.

[92] P. Ratnasingham. The importance of trust in electronic commerce. *Electronic Networking Applications and Policy*, Vol. 8:313–321, 1998.

[93] D. Reyes, C. A. M. Festin, and S. Pancho-Festin. Incentive-based self-organized public key management for mobile ad hoc networks. In *Proceedings of 3rd IEEE International Conference on Network and System Security*, pages 565–570, 2009.

[94] T. Saito, R. Hatsugai, and T. Kito. On compromising password-based authentication over https. In *Proceedings of 20th IEEE International Conference on Advanced Information Networking and Applications*, pages 869–874, 2006.

[95] R. M. Savola. Node level security management and authentication in mobile ad hoc networks. In *Proceedings of 10th IEEE International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 449–458, 2009.

[96] J.-M. Seigneur. *Trust, Security and Privacy in Global Computing.* PhD thesis, Trinity College Dublin, 2005.

[97] A. Singh and L. Liu. Trustme: Anonymous management of trust relationships in decentralized p2p systems. In *Proceedings of IEEE International Conference on Peer-to-Peer Computing*, pages 142–149, 2003.

[98] A. Sjoholm, L. Seitz, and B. Sadighi. Secure communication for ad-hoc, federated groups. In *Proceedings of the 7th ACM symposium on Identity and trust on the Internet*, pages 48–58, 2008.

[99] M. Steiner, G. Tsudik, and M. Waidner. Key agreement in dynamic peer groups. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11:769–780, 2000.

[100] Y. L. Sun, W. Yu, Z. Han, and K. J. R. Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, Vol. 24:305–317, 2006.

[101] C. Tartary and H. Wang. Combining prediction hashing and mds codes for efficient multicast stream authentication. In *Proceedings of the 12th Springer Australasian conference on Information security and privacy*, pages 293–307, 2007.

[102] N. M. Thamrin, G. Witjaksono, A. Nuruddin, and M. S. Abdullah. An enhanced hardware-based hybrid random number generator for cryptosystem. In *Proceedings of the IEEE International Conference on Information Management and Engineering*, pages 152–156, 2009.

[103] G. Theodorakopoulos and J. Baras. Trust evaluation in ad-hoc networks. In *Proceedings of ACM workshop on Wireless security*, pages 1–10, 2004.

[104] J. van der Merwe, D. Dawoud, and S. McDonald. Fully self-organized peer-to-peer key management for mobile ad hoc networks. In *Proceedings of the 4th ACM workshop on Wireless security*, pages 21–30, 2005.

[105] J. Viega, T. Kohno, and B. Potter. Trust (and mistrust) in secure applications. *Communications of the ACM*, Vol. 44:31–36, 2001.

[106] M. Virendra and S. Upadhyaya. Securing information through trust management in wireless networks. In *Proceedings of Workshop on Secure Knowledge Management*, 2004.

[107] A. von Bidder and N. Weiler. Key exchange (kx) - a next generation protocol to synchronise pgp keyservers. In *Proceedings of 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 249–254, 2003.

[108] D. Wang. An energy-efficient clusterhead assignment scheme for hierarchical wireless sensor networks. *Springer International Journal of Wireless Information Networks*, Vol. 15:61–71, 2008.

[109] G. Wang, Q. Wang, J. Cao, and M. Guo. An effective trust establishment scheme for authentication in mobile ad hoc networks. In *Proceedings of 7th IEEE International Conference on Computer and Information Technology*, pages 749–754, 2007.

[110] J. Wang, S. Lian, and G. Liu. On the analysis and design of secure multimedia authentication scheme. In *Proceedings of 3rd IEEE International Conference on Communications and Networking in China*, pages 1298–1302, 2008.

[111] X. Wang, F. Dai, L. Qian, and H. Dong. A way to solve the threat of selfish and malicious nodes for ad hoc networks. In *Proceedings of IEEE International Symposium on Information Science and Engineering*, pages 369–370, 2008.

[112] Y. Wang, V. C. Giruka, and M. Singhal. Truthful multipath routing for ad hoc networks with selfish nodes. *Elsevier Journal of Parallel and Distributed Computing*, Vol. 68:778–789, 2008.

[113] Z. Wang and C. Chigan. Countermeasure uncooperative behavior with dynamic trust-token in vanets. In *Proceedings of IEEE International Conference on Communications*, pages 3959–3964, 2007.

[114] A. Weimerskirch and D. Westhoff. Identity certified authentication for ad-hoc networks. In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 33–40, 2003.

[115] C. Xiao, S. Ma, J. Niu, L. Wang, B. Shan, and T. Chen. A novel security scheme for video conference system with wireless terminals. In *Proceedings of 5th IEEE International Symposium on Embedded Computing*, pages 101–106, 2008.

[116] S. Xu. On the security of group communication schemes based on symmetric key cryptosystems. In *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 22–31, 2005.

[117] Z. Yan, P. Zhang, and T. Virtanen. Trust evaluation based security solution in ad hoc networks. Technical report, Nokia Research Center, Helsinki, Finland, 2003.

[118] Y. Zhang, W. Lee, and Y.-A. Huang. Intrusion detection techniques for mobile wireless networks. *Wireless Networks*, Vol. 9:545–556, 2003.

[119] Y. Zhang, W. Liu, W. Lou, and Y. Fang. Securing mobile ad hoc networks with certificateless public keys. *IEEE Transactions on Dependable and Secure Computing*, Vol. 3:386–399, 2006.

[120] Q. Zheng, X. Hong, J. Liu, and L. Tang. A secure data transmission scheme for mobile ad hoc networks. In *Proceedings of IEEE Global Telecommunications Conference*, pages 1006–1010, 2007.

[121] S. Zhong, J. Chen, and Y. R. Yang. Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of 22nd Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 1987–1997, 2003.

[122] S. Zhong, T. M. Khoshgoftaar, and S. V. Nath. A clustering approach to wireless network intrusion detection. In *Proceedings of 17th IEEE International Conference on Tools with Artificial Intelligence*, 2005.

[123] Z. Zhou, G. Xu, J. He, J. Jiang, and C. Deng. Research of secure anycast group management. In *Proceedings of 4th IEEE International Conference on Networked Computing and Advanced Information Management*, pages 604–608, 2008.

[124] H. Zhu, F. Bao, and R. H. Deng. Computing of trust in wireless networks. In *Proceedings of 60th IEEE Vehicular Technology Conference*, pages 2621–2624, 2004.