Security On Multicast In Mobile Ad Hoc Networks

A Project Report Submitted in the partial fulfillment of the requirements for the award of degree of

Master of Technology
in
Computer Science

By
Ashok Kumar Sama

Department of Computer and Information Sciences
University of Hyderabad
Hyderabad, India

April 2010

CERTIFICATE

This is to certify that the project work entitled "**Security On Multicast In Mobile Ad Hoc Networks**" being submitted to University of Hyderabad by **Ashok Kumar Sama** (Reg. No. 08MCMT31), in partial fulfillment for the award of the degree of Master of Technology in Computer Science, is a bonafide work carried out by him under my supervision.

Mr. Wilson Naik
Project Supervisor,
Department of CIS,
University of Hyderabad

Head of Department,
Department of CIS,
University of Hyderabad

Dean,
School of MCIS,
University of Hyderabad

To,

My Parents and Friends.

Acknowledgments

I would like to express my sincere gratitude to **Mr. Wilson Naik**, my project supervisor, for valuable suggestions and keen personal interest throughout the progress of my course of research. He encouraged, supported, corrected and guided me during the project. The project has been a learning and growing experience for me. It is a good piece of work done in his guidance and a great opportunity to learn many things.

I am thankful to my lab mate **N. Srinivas Naik** for their advice and discussions with me. I would also like to thank the **Open Source Community** who provided the free software and documentation to work with.

I am extremely grateful to our Head of the Department, **Prof. Arun Agarwal**, for providing excellent computing facilities and a nice atmosphere for doing my project.

I would like to take this opportunity to thank my friends who have been morale boosters for me, encouraged me to take up this course and supported me throughout this course with their love and affection. My special thanks to my wonderful parents who have always supported me in all my decisions.

Ashok Kumar Sama.

Abstract

A mobile adhoc network (MANET) is a self organizing system with multiple mobile nodes. Nodes communicate among themselves via wireless links with no fixed infrastructure or centralized administration such as base station or access points.

Nodes in a MANETs operate both as host as well as routers to forward packets for each other in a multi-hop fashion.There are many applications in wired and wireless networks. By multicasting, a single message can be delivered to multiple receivers simultaneously. It greatly reduces the transmission cost when sending the same packet to multiple recipients.

The security issue of MANETs in group communications is more challenging because of dynamicity and involvement of multiple senders and receivers. In multicasting, mobile adhoc networks are prone to different types of attacks from malicious nodes due to several reasons. Some of these attacks are Rushing attack, Blackhole attack, Sybil attack, Neighbor attack, Impersonation and Jellyfish attack. To combat such attacks from the malicious nodes an efficient authentication protocol is required. we try to investigate Hop-by-Hop, efficient authentication protocol, called HEAP to prevent the multicast attacks of ODMRP protocol and we found Hop-by-Hop, efficient authentication protocol in ODMRP is efficient compared to other schemes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is Mobile Adhoc Network

A mobile adhoc network is a self-organizing system of mobile nodes that communicate with each other via wireless links with no fixed infrastructure or centralized administration such as base stations or access points. Nodes in a MANET operate both as hosts as well as routers to forward packets for each other in a multi-hop fashion. MANETs are suitable for applications in which no infrastructure exists such as military battlefield, emergency rescue, vehicular communications and mining operations.In these applications, communication and collaboration among a given group of nodes are necessary.Instead of using multiple unicast transmissions, it is advantageous to use multicast in order to save network bandwidth and resources, since a single message can be delivered to multiple receivers simultaneously.

Existing multicast routing protocols in MANETs can be classified into two categories: tree-based and mesh-based. In a multicast routing tree, there is usually only one single path between a sender and a receiver, while in a routing mesh, there may be multiple paths between each sender receiver pair. Routing meshes are thus more suitable than routing trees for systems with frequently changing topology such as MANETs due to the availability of multiple paths between a source and a destination. Multicast data may still be delivered to the destination on alternative paths even when the main route breaks. Example tree-based

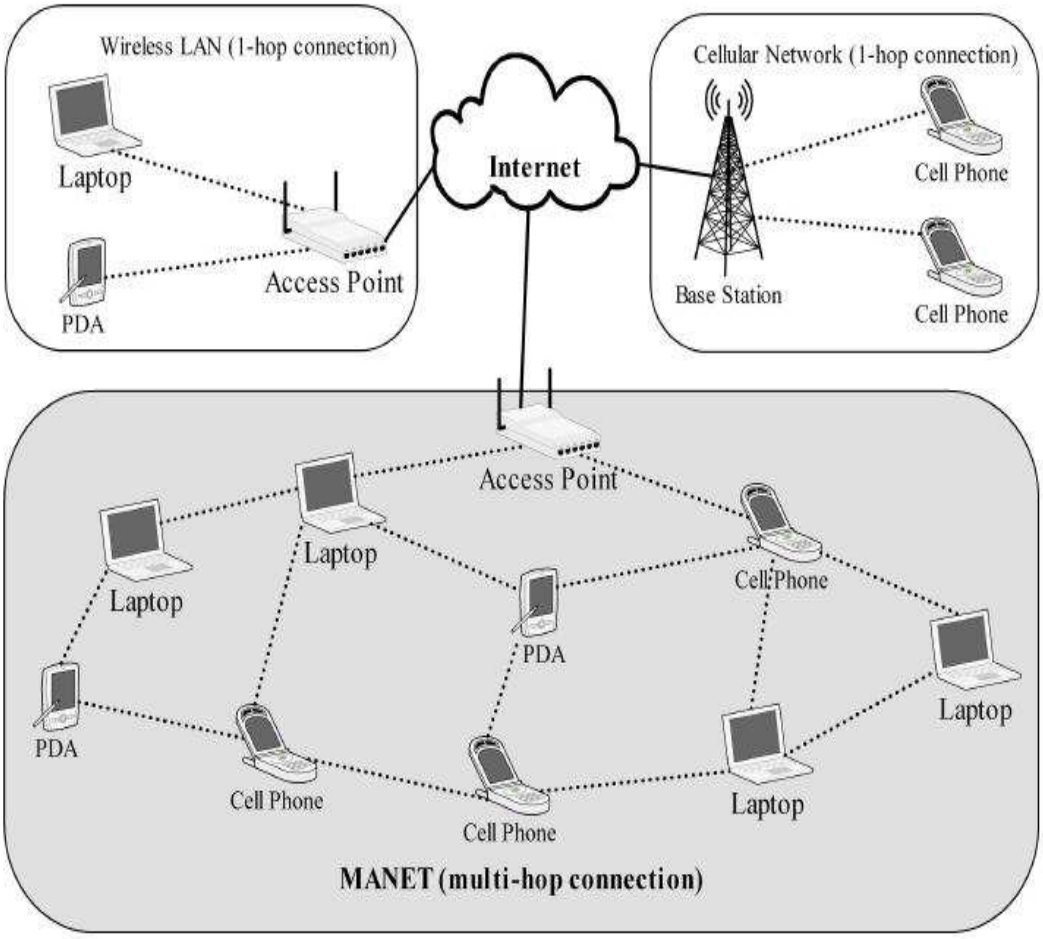

Figure 1.1: Mobile ad hoc networks

multicast routing protocols are MAODV, AMRIS, BEMRP and ADMR. Typical mesh-based multicast routing protocols are ODMRP, FGMP, CAMP, DCMP, and NSMP.

Among all the research issues, security is an essential requirement in MANET environments. Compared to wired networks, MANETs are more vulnerable to security attacks due to the lack of a trusted centralized authority, lack of trust relationships between mobile nodes, easy eavesdropping because of shared wireless medium, dynamic network topology, low bandwidth, and battery and memory constraints of mobile devices.

1.2 Manet Security

In MANETs, security is one of the most important concerns because a MANET system is much more vulnerable to attacks than a wired or infrastructure-based wireless network. Designing an effective security protocol for MANETs is a very challenging task. This is mainly due to the unique characteristics of MANETs, namely shared broadcast radio channel, insecure operating environment, lack of central authority, lack of association among users, limited availability of resources, and physical vulnerability.

Security issues of MANETs in group (multicast) communications are even more challenging because of the involvement of multiple senders and multiple receivers. Although several types of security attacks in MANETs have been studied in the literature, the focus of earlier research is only on unicast (point-to-point) applications. The impacts of security attacks on multicast in MANETs have not yet been explored.

1.2.1 Attacks

We present a simulation-based study of the effects of different types of attacks on mesh-based multicast in MANETs. We consider the most common types of attacks, namely rushing attack, blackhole attack, neighbor attack and jellyfish attack.

- **Rushing attack:** Many demand-driven protocols such as ODMRP, MAODV, FGMP, and ADMR, which use some form of duplicate suppression in their operations, are vulnerable to rushing attacks. When source nodes flood the network with route discovery packets in order to find routes to the destinations, each intermediate node processes only the first non-duplicate packet and discards any duplicate packets that arrive at a later time. A rushing attacker exploits this duplicate suppression mechanism by quickly forwarding route discovery packets in order to gain access to the forwarding group. Rushing attacks were first introduced by Hu et al. in [11].

- **Blackhole attack:** A blackhole attacker first needs to invade into the multicast forwarding group, for example by implementing rushing attack, in order to intercept data packets of the multicast session. It then drops some or all data packets it receives instead of forwarding them to the next node on the routing path. This type of attack often results in very low packet delivery ratio.

- **Neighbor attack:** Upon receiving a packet, an intermediate node records its ID in the packet before forwarding the packet to the next node. An attacker, however, simply

forwards the packet without recording its ID in the packet to make two nodes that are not within the communication range of each other believe that they are neighbors (i.e., one-hop away from each other), resulting in a disrupted route.

- **Jellyfish attack:** A jellyfish attacker first needs to intrude into the multicast forwarding group. It then delays data packets unnecessarily for some amount of time before forwarding them. This results in significantly high end-to-end delay and thus degrades the performance of real-time applications. Jellyfish attacks in MANETs were first discussed in [1].

1.2.2 Threats and vulnerabilities

The threats and vulnerabilities of MANETs are :

- **Attacks due to vulnerability of wireless links :**These include passive eavesdropping, active impersonation, message replay, message distortion, injection of erroneous messages, congestion or denial of service, etc.

- **Attacks due to compromised nodes :** Poor physical protection of nodes in a hostile environment results in a non-negligible probability of a node being compromised. Attacks launched from within the network by compromised nodes should be taken into consideration. The security mechanisms should not be based on a central entity to avoid single point(s) of failure in the network. It is also important to ensure that the system fails gracefully. In general, the failure of a single node should not jeopardize the security of the system as a whole.

- **Attacks due to vulnerability of security mechanisms :** These include malicious replacement of public keys, compromise of keys, etc. The security mechanisms should be resistant to both passive and active adversaries. Recently, there have been a number of side-channel attacks on many popular cryptosystems. These attacks do not break the security mechanism as such, but use information like padding in messages, key-strokes, power analysis, etc to recover the key. The key recovery is not due to the weaknesses in the security mechanism. Examples include electromagnetic and differential power analysis techniques to break smart-card security and padding attacks on Cipher-block-chaining mode of encryption.

- **Denial of Service threats:** A Denial of Service (DoS) can be created by malicious nodes or by unintentional failure of node(s) in the network. Forms of DoS vary from

a simple flooding of a resource to congesting parts of the network to reconfiguration of routing protocol by malicious nodes leading to Byzantine failures.

1.2.3 Security aspects in ad hoc networks

The security requirements depend largely on the mission and operating conditions of the MANET. For military applications, the requirements will be stringent in terms of confidentiality and resistance to DoS attacks. For example, we might have to ensure that wireless traffic does not reveal the location of a target to the enemy in a military mobile ad hoc network while it is not necessary to do so in a civilian ad hoc network. The following are the general security requirements in a MANET [1, 3, 4]. Depending on the application, the constraints and requirements would eliminate or make more critical one or more of the following or have added restrictions.

- **Authentication :**The receiver of a message must be able to verify the identity of the originator of the message, thereby preventing an intruder from masquerading as a legitimate source of the message.
- **Integrity :** The receiver of a message should be able to check whether a message was modified in transit. This is to avoid accepting packets that have been modified by a hostile node, while in transit.
- **Confidentiality :** Confidentiality is needed to ensure that certain information (like tactical military information or routing information that might reveal the location of the target) is never disclosed to unauthorized entities. Confidentiality is required only when the message is to be kept secret.
- **Non-repudiation:**The originator of the message cannot deny having sent the message. Non-repudiation is useful for detection and isolation of compromised nodes.
- **Availability :** Availability ensures functionality of network services despite DoS attacks.The security mechanisms should not make any assumptions about the availability of specific nodes at any given time. Nodes may be idle, shut down for a while or could be compromised.
- **Access Control :** Some systems may require that only certain nodes be entitled to perform certain tasks, e.g posting certain messages (commands/orders), revoking a certain policy / trust information, etc. Access control mechanisms must be established to deal with these issues.

- **Scalability :**Security mechanisms may be required to be scalable to handle a large number of nodes, depending on the application.

- **Adaptability to changes :**Frequent changes in topology and membership make the ad hoc network dynamic. Trust relations among nodes also change with time and so does a nodes affiliation with an administrative domain. The security mechanisms should not have a static configuration and should be able to adapt to changes on the fly.

- **Key Management :** The only way to accurately enforce authentication, integrity and non-repudiation is by using some form of cryptography, which requires the distribution/exchange of encryption/authentication key information among message senders and receivers. A key management module consists of a trust model, a cryptosystem, key establishment module, a key storage and key distribution service. Some group-oriented applications may require key revocation services as well.

1.3 Security mechanisms

A variety of security mechanisms have been invented to counter malicious attacks. The conventional approaches such as authentication, access control, encryption, and digital signature provide a first line of defense. As a second line of defense, in- trusion detection systems and cooperation enforcement mechanisms implemented in MANET can also help to defend against attacks or enforce cooperation, reducing selfish node behavior.

- **Preventive mechanism:**The conventional authentication and encryption schemes are based on cryptography, which includes asymmetric and symmetric cryptography. Cryptographic primitives such as hash functions (message digests) can be used to enhance data integrity in transmission as well. Threshold cryptography can be used to hide data by dividing it into a number of shares. Digital signatures can be used to achieve data integrity and authentication services as well.It is also necessary to consider the physical safety of mobile devices, since the hosts are normally small devices, which are physically vulnerable. For example, a device could easily be stolen, lost, or damaged. In the battlefield they are at risk of being hijacked. The protection of the sensitive data on a physical device can be enforced by some security modules, such as tokens or a smart card that is accessible through PIN, passphrases, or biometrics. Although all of these cryptographic primitives combined can prevent most

attacks in theory, in reality, due to the design, implementation, or selection of protocols and physical device restrictions, there are still a number of malicious attacks bypassing prevention mechanisms.

- **Reactive mechanism:**An intrusion detection system is a second line of defense. There are widely used to detect misuse and anomalies. A misuse detection system attempts to define improper behavior based on the patterns of well-known attacks, but it lacks the ability to detect any attacks that were not considered during the creation of the patterns; Anomaly detection attempts to define normal or expected behavior statistically. It collects data from legitimate user behavior over a period of time, and then statistical test are applied to determine anomalous behavior with a high level of confidence. In practice, both approaches can be combined to be more effective against attacks. Some intrusion detection systems for MANET have been proposed in recent research papers.

1.4 Motivation

Mobile Ad Hoc networks (MANETs) are vulnerable due to its fundamental characteristics,such as open medium, dynamic topology, distributed operation and constrained capability. Security is a central requirement for mobile Ad Hoc networks.

- Previous security research focused only on unicast (one sender to one receiver) communications.
- Research is going on the vulnerabilities of multicast communications yet.
- Multicast is more complicated due to the involvement of multiple senders and multiple receivers.
- There are some secure protocols for multicast.

Security and robustness will impact the design of the standard for Ad Hoc networks is the main motivation for this thesis.

1.5 Outline of the Thesis

The meaning part of the thesis is organized as follows. In Chapter 1, we present the introduction of mobile adhoc networks, different attacks of the manets and their security

mechanisms. In Chapter 2, we present a literature review of multicast routing in manets, various attacks of the manets and the solutions of these attacks. In chapter 3, we present the ODMRP overview and its operation. In chapter 4, we present HEAP:Hop-by-Hop Efficient Authentication protocol and its implementation and Chapter 5 summarizes our thesis and outlines our future work.

Chapter 2

Literature Survey

In this chapter, we discuss Different types of attacks and solutions of the attacks of multicast routing in MANETs and previous security studies for MANETs are presented.

2.1 Multicast Routing in Mobile Ad Hoc Networks

The security issues of Mobile Ad-hoc Networks (MANETs) are more challenging in a multicasting environment with multiple senders and receivers. There are different kinds of attacks by malicious nodes that can harm a network and make it unreliable for communication. These attacks can be classified as passive and active attacks.

2.2 Types of security attacks

There are two types of security attacks:

1. Passive Attacks
2. Active Attacks

In a passive attack, a malicious node either ignores operations supposed to be accomplished by it (examples:silent discard, partial routing information hiding), or listens to the channel, attempting to retrieve valuable information (example: eavesdropping). In a active attack, information is inserted to the network and thus the network operation or some nodes may be harmed.Examples are impersonation/spoofing,modification, fabrication and disclosure

attack. Active attacks include modification, fabrication, re-play and deletion of packets. Each of these attacks is concisely explained as follows.

- **Packet modification:** An attacker illegally changes contents of packets that it receives or overhears and inserts them into the network.
- **Packet fabrication:** An attacker generates bogus packets and sends them to the network. It can either use its identity or spoof other nodes identity.
- **Packet replay:** An attacker stores packets that it receives or overhears and sends them to the network later.
- **Packet deletion:** An attacker drops all or part of packets that it should forward.

Depending on the types of packets on which attacks are conducted, attacks can be classified into data attacks and control attacks.

In data attacks, attackers perform passive/active attacks on data packets. Active attacks on data packets are denial-of-service attacks in general. Packet modification, fabrication and replay attacks are easily detected by using message authentication codes and mechanisms such as timestamps but these attacks consume network resources unnecessarily.Packet deletion attacks are difficult to detect and disrupt normal packet delivery.

In control attacks, attackers perform passive/active attacks on control packets such as multicast routing packets. Active attacks on control packets try to illegally modify multicast delivery trees.Attacks can be again classified into outsider attacks and insider attacks.Attacks conducted by unauthorized nodes in an ad hoc network are called outsider attacks and are in general easy to detect. Attacks launched by authorized nodes that are captured by enemies and, therefore,compromised are called insider attacks and are difficult to verify. The following two tables, precisely table 2.1 summarizes the attacks and table 2.2 represents the solutions in each layer in MANET.

Layer	**Attacks**
Application layer	Repudiation, data corruption
Transport layer	Session hijacking, SYN flooding
Network layer	Wormhole, blackhole, Byzantine, flooding, resource consumption
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11), WEP weakness
Physical layer	Jamming, interceptions, eavesdropping

Table 2.1: Security Attacks on each layer in MANET

Layer	Security Issues
Application layer	Detecting and preventing viruses, worms,malicious codes, and application abuses
Transport layer	Authentication and securing end-to-end or point-to-point communication through data encryption
Network layer	Protecting the ad hoc routing and forwarding protocols
Data link layer	Protecting the wireless MAC protocol and providing link layer security support
Physical layer	Preventing signal jamming denial-of-serviceattacks

Table 2.2: Security Solutions for MANET

2.2.1 Security threats and countermeasures

The following table summarizes the potential security attacks and the actions that can be taken to prevent the attacks.

Layers	Attacks	Solutions
Application layer	Lack of cooperation attacks, Malicious code attacks (virus,worms, spywares, Trojan horses) etc.	Cooperation enforcement(Nuglets, Confidant, CORE) . mechanisms, Firewalls, IDS etc.
Transport layer	Session hijacking attack, SYN flooding attack, TCP ACK storm attack etc.	Authentication and securing end-to-end or point-to-point communication, use of public cryptography (SSL, TLS, SET, PCT) etc.
Network layer	Routing protocol attacks (e.g.DSR, AODV etc.), cache poisoning, table overflow attacks, Wormhole, blackhole, Byzantine, flooding, resource consumption, impersonation, location disclosure attacks etc.	Source authentication and message integrity mechanisms to prevent routing message modification, Securing routing protocols (e.g. IPSec, ESP, SAR, ARAN) to overcome blackhole, impersonation attacks, packet leashes, SECTOR mechanism for wormhole attack etc.
Data link layer	Traffic analysis, monitoring, disruption MAC (802.11), WEP weakness etc.	No effective mechanism to prevent traffic analysis and monitoring, secure link layer protocol like LLSP, using WPA etc
Physical layer	Jamming, interceptions eavesdropping	Using Spread spectrum mechanisms e.g. FHSS, DSSS etc.

Table 2.3: Security threats and countermeasures

In MANETs, communication and collaboration among a given group of nodes are usually necessary. Multicast routing is a preferred communication mechanism over multiple unicast transmissions in order to deliver the same data to multiple recipients. Multicast is a form of communication that delivers information from a source to a set of destinations simultaneously in an ecient manner; the messages are delivered over each link of the network only once and only duplicated at branch points, where the links to the destinations split (Figure 2.1). Multicast communications are especially useful in applications such as distribution of newsletters, software and stock quotes; audio/video conferencing; online education; online Internet games; and communications among military troops or rescue teams.

Existing multicast routing protocols in mobile ad-hoc networks can be classied broadly into two categories: tree-based and mesh-based. In tree-based multicast protocols, there is usually only one single path between a sender and a receiver, while in mesh-based multicast protocols, there may be multiple paths between each sender-receiver pair. Example tree-based multicast protocols are MAODV, AMRIS, BEMRP and ADMR. Typical mesh-based multicast protocols are PMR, ODMRP, FGMP, CAMP, DCMP and NSMP. AMRoute

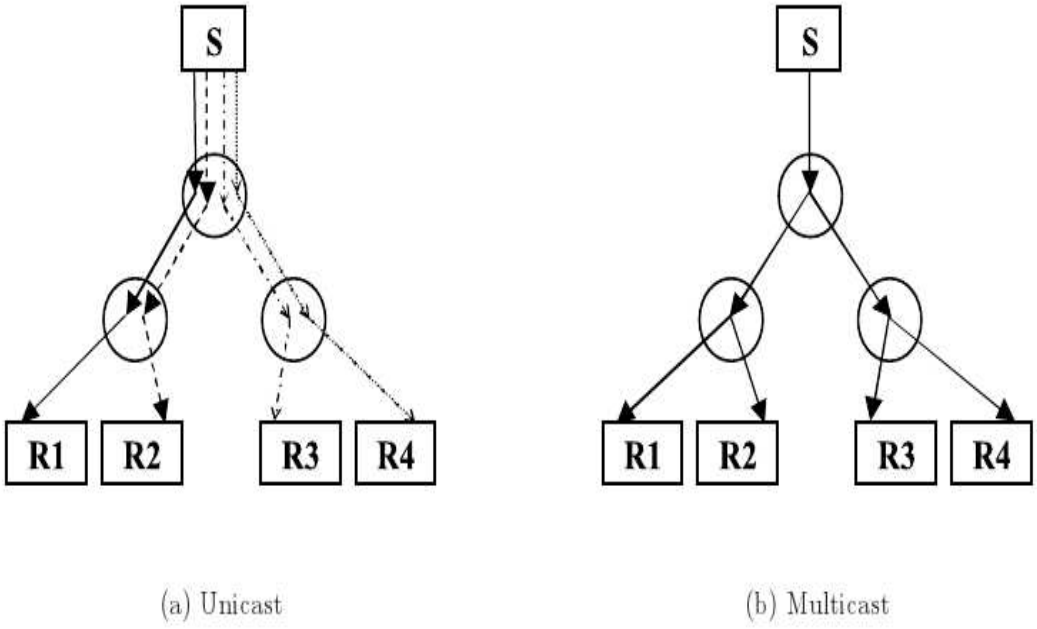

Figure 2.1: Multicast versus unicast

and MCEDAR are hybrid multicast protocols that provide both mesh-based and tree-based infrastructure.

Compared to tree-based protocols, mesh-based protocols are more robust and suitable for systems with frequently changing topology such as MANETs. Maintaining a single multicast tree is not appropriate for MANETs because the tree could easily break due to highly dynamic topology and node mobility. The instability of multicast trees results in higher packet losses, and an increase in the number of retransmissions. In contrast, routing meshes allow multicast data to be delivered to a destination on alternative paths even when the main route breaks, due to the availability of multiple paths between a source and a destination.

In our study, we used the On-Demand Multicast Routing Protocol (ODMRP)[1], a mesh-based routing protocol, due to its simple implementation and high packet delivery ratio. The advantages of ODMRP make it a very popular multicast routing protocol for mobile ad hoc networks: ODMRP has been cited, evaluated and compared with other multicast routing protocols in over 50 research projects and papers. The following sections describe the operation of ODMRP in detail.

2.3 Defending Against Outsider Attacks

2.3.1 Introduction

MANETs must provide various levels of security guarantees to dierent applications for their successful deployment and usage. However, due to their wireless links and lack of central administration, MANETs have far greater security concerns than conventional networks . For example, it is easy for attackers to enter or leave a wireless network and eavesdrop since no physical connection is required. Without a security scheme in place, an outsider node can easily pretend to be an insider and participate in routing packets. Therefore, it can directly attack the network by dropping packets, tampering with packets, injecting false packets or ooding the network. As a result, it is possible to launch sophisticated wormhole, man-in-the-middle and Denial of Service (DoS) attacks with ease, or to impersonate another node.

To combat such attacks, we study hop-by-hop packet authentication as a fundamental security strategy to block attacks from outsiders at the rst hop. We propose a Hop-by-hop, Ecient Authentication Protocol, called HEAP, which is suitable for multicast, unicast or broadcast applications. In essence, HEAP uses a modied HMAC [2] based algorithm that utilizes two keys. Specically, in the initial bootstrapping phase every node (i) shares a pairwise secret hash key, called okey, with each of its neighbors, and (ii) generates one common secret hash key, called ikey, and securely distributes it to all of its one-hop neighbors. Now consider the case where a node wants to broadcast a message to all of its neighbors. The naive way would be to generate a new MAC for every individual neighbor using its pairwise key. However, we present a more ecient way of generating MACs. We reexamined the working principles of the HMAC algorithm [2] and divided the HMAC computation into two steps so that we could signicantly lower the computational cost. In our scheme, we pay special attention to the eciency of the proposed authentication scheme since the nodes in MANETs are often constrained in battery power, memory, bandwidth, and CPU power. We explain the details of HEAP in Section 4.

2.3.2 Related Work

MANETs security research includes areas such as intrusion detection, secure routing, key establishment and distribution, and authentication. Intrusion detection and response is addressed in various studies. In [3] Dahill et al identify several security vulnerabilities in the

popular routing protocols such as AODV and DSR. They propose to use asymmetric cryptography to secure these protocols. Other secure routing protocols are presented in [4], but they assume the pre-existence and pre-sharing of public and secret keys for all initial members. The key sharing problem is considered by researchers in [1]. Works such as [5] aim at designing self securing MANETs without needing third party certifying authorities outside the network. Other researchers have proposed schemes, including SEAD and Ariadne, for securing routing protocols. Most of these schemes authenticate only control packets and not data packets. Extending these schemes to directly authenticate data packets would result in too much overhead. On the other hand, not authenticating data packets leads to serious vulnerabilities against attacks such as DoS, replay, man in the middle, wormhole, and impersonation attacks. Since it is important to guard against these attacks we need to authenticate data as well as control packets. We have designed our scheme, HEAP, to authenticate both types of packets.

For comparison, three other packet authentication schemes presented in the literature are also designed to authenticate data as well as control packets: TESLA [6], LHAP [4, 7] and Lu and Poochs algorithm [8] (hereafter referred to as Lu). These three algorithms approach the problem of authentication in slightly different ways but all of them use one-way hash key chains. For example, TESLA provides end-to-end authentication. Although it is not specically designed for MANETs, it can easily be applied to them. LHAP and Lus algorithm build on the principles of TESLA and are specically intended for MANETs. They provide hop-by-hop authentication just like HEAP. For this reason, we found it suitable to compare the security and performance of our algorithm against theirs. In the rest of this section, we briey describe these schemes and discuss their major security merits and vulnerabilities.

2.3.3 TESLA

TESLA is a very efficient multicast stream authentication protocol that can be applied towards wired as well as wireless networks. It can be used for unicast and broadcast applications as well, and its parameters can be adjusted to adapt to varying network conditions, such as congestion. However, since TESLA was not specically designed for MANETs but for LANs in general, it has certain drawbacks which make it less suitable for MANETs. For instance, packets are not authenticated at every hop, instead they are only authenticated by the nal receiver after a delay of several seconds. The packets are held in a cache at the receiving node until the hash key used to authenticate them has been disclosed by the

sender. If the hash key can regenerate the MAC of the packet, the packet is considered authentic; otherwise, it is dropped. Intermediate nodes simply forward the packets without authentication. As a result, an attacker can easily launch Denial of Service (DoS) attacks by ooding the network with bogus packets. These packets can ood the packetcaches of all receiving nodes causing them to overow and drop legitimate packets. An attacker could also easily launch worms that would quickly propagate throughout the network unchecked by the forwarding nodes. Hop-by-Hop authentication restricts the damage of this type of attack by limiting its propagation.

In addition, TESLA requires the clocks of all the nodes in the network to be loosely synchronized. This requires a secure time synchronization protocol and possibly time servers to synchronize with. Such time servers may not always be available or reachable in a MANET. Any vulnerability in the synchronization mechanism could potentially allow an attacker to compromise the scheme and successfully send forged packets. A secure time synchronization protocol suitable for MANETs needs to be developed and tested for TESLA to be applicable in MANETs.

2.3.4 LHAP

LHAP was specically designed for MANETs and it introduces the idea of hop-by-hop authentication. It builds on the principles of TESLA and tries to overcome some of its drawbacks. For example, LHAP does not require loose time synchronization. It is also very ecient and it authenticates packets instantly, reducing latency and eliminating the need for a cache at every node. However, LHAP is vulnerable to wormhole and man-in-the-middle attacks. LHAPs authors themselves point out this vulnerability. Briey speaking, this vulnerability allows an outside attacker to eavesdrop on authentic packets sent by a node, modify them and retransmit them to another node in the network that is outside the range of the original node. This can be done using a dedicated private channel between two colluding nodes (i.e., a wormhole) or by the attacker being in the middle of the sender and the receiver that are outside each others range.

To thwart this attack, LHAPs authors suggest using GPS devices in all the nodes to ascertain whether a sending node should be within transmission range of the receiving node. We do not think it is practical to equip all the nodes with GPS devices, especially if those nodes are sensors or small wireless devices. We stress that even with GPS coordinates, it is very dicult to gure out whether another node is within transmission range because

the range is affected by complex factors such as terrain, weather conditions, radio noise, and transmission power level. In a MANET, all of these factors may continually change and so it is not possible to verify whether a node is within transmission range, unless it is unambiguously too far off or very close. LHAPs authors admit that their scheme does not fully address this attack.

Because of this vulnerability an attacker can successfully send forged packets which will be considered authentic by the receiver. This goes against the purpose of the scheme which is to authenticate all incoming packets and prevent unauthentic packets from being propagated, leaving LHAP potentially vulnerable to wormhole, man in the middle, DoS and replay attacks.

2.3.5 Lu and Pooch

Lu and Poochs algorithm builds on LHAP. Like LHAP, Lus algorithm also uses hop-by-hop authentication and is ecient, but unlike LHAP it uses only one key at every node instead of two. Like TESLA, Lu also uses delayed key disclosure causing network latency to increase, but the delay is not as large as TESLAs. We show here that Lus algorithm is also vulnerable to wormhole and man-in-the-middle attack. Delayed key disclosure schemes like TESLA and Lu have a security condition that a data packet can only be considered safe and authentic if the receiver receives the packet before the sender discloses the corresponding key used to generate the MAC for that packet. Otherwise, an adversary would retrieve the disclosed key, generate the MAC for a forged packet and send it to the receiver. The receiver would consider that packet to be authentic since it would not know that the key had already been disclosed.

To guard against this possibility, the sender in TESLA informs the receiver at what time it will transmit the key disclosure packet. If a data packet is received well before this time and its MAC is later authenticated with the disclosed key, the packet is considered authentic. But if a packet is received after the key disclosure time, then it is dropped even if the receiver has not yet received the key. This scheme requires that the clocks of the sender and the receiver must be synchronized, which is dicult to achieve. To circumvent this problem of synchronization, Lu introduced an idea follows. Instead of the sender stating that the next key update packet will be broadcast at time t, the sender will state in the data packet a delay parameter that will state that the key will be disclosed d milliseconds after the transmission of this data packet.

But there is a problem with this latter approach. A clever attacker can launch a wormhole or man-in-the-middle attack where it will forward all the packets from the sender to the receiver during the bootstrap phase so that the receiver mistakenly believes that the sender is its neighbor. Then the attacker will wait until it receives a key update packet from the sender. Once the key is received by the attacker, the attacker can forge many packets and send them to the receiver, and then disclose the key. Because the receiver does not expect a key update packet from the sender at a xed time t, it has no way of knowing that this key was transmitted by the sender some time ago and is now obsolete. The attacker can use this obsolete key and all subsequent keys to successfully transmit forged packets. In this way the attacker can thwart Lus scheme.

2.4 Outsider vs. Insider nodes

An outsider node is a node that is not an authorized member of the MANET whereas an insider node is an authorized member. For instance, in a military setting each authorized soldier might possess a signed certicate from a trusted third party granting him membership in the MANET. Such a node is an insider node. Any node not possessing such a certicate or possessing a revoked certicate is considered an outsider node.

Detecting the attacks from insiders is left to the Intrusion Detection Systems. However, we do provide a foundation on which a response system can be based by providing the capability to eectively cut o a compromised insider from the MANET. In addition, HEAP oers some level of protection against insiders who try to forge packets and impersonate other insiders. But because insiders already have access to the MANET, it is easy for them to launch more sophisticated attacks rather than simply trying to forge packets.
We now list the keys security goals in defending the underlying network against outsider nodes.

1. Any packet transmitted by an outsider node should be immediately dropped by the receiving insider node at the rst hop with a very high probability. In other words, packets sent by outsiders should not be allowed to propagate through the MANET.
   By fullling this requirement, we can successfully guard against a myriad of attacks by the outsider, such as DoS attacks that attempt to ood the network, wormhole attacks, man-in-the-middle, SYN ooding etc. This is because we are effectively disabling the outsiders ability to route any packets to any node that is not its neighbor. Even a neighboring node will simply drop packets from the outsider. However, this require-

ment dictates that we authenticate every packet at every hop, which in turn means that the authentication mechanism should be extremely efficient.

2. The outsider node is assumed to have the capability to spoof its identity, such as spoong its IP and MAC addresses to impersonate an insider node. We cannot rely on these markers to verify the origin of a packet. the origin of a packet.

3. The outsider is assumed to have access to the wireless channel so it can eavesdrop on legitimate traffic. If the trac is supposed to remain condential, end-to-end encryption should be used to protect it. However, we assume that the encrypted trac and any associated MAC tags are visible to the outsider.

4. If a third party intrusion detection system (IDS) discovers an insider to be compromised, we must be able to exclude that insider from propagating any more packets within the MANET. Certicate revocation lists (CRL) may be used to revoke the certicates of compromised insiders.

Chapter 3

ODMRP Overview

3.1 Introduction

ODMRP applies ”on-demand” routing techniques to avoid channel overhead and improve scalability. It uses the concept of ”forwarding group,” a set of nodes responsible for forwarding multicast data, to build a forwarding mesh for each multicast group. By maintaining and using a mesh instead of a tree, the drawbacks of multicast trees in mobile wireless networks (e.g., intermittent connectivity, traffic concentration, frequent tree reconfiguration, non-shortest path in a shared tree, etc.) are avoided. A soft-state approach is taken to maintain multicast group members. No explicit control message is required to leave the group. We believe the reduction of channel/storage overhead and the relaxed connectivity make ODMRP more attractive in mobile wireless networks[9].

Properties of ODMRP :

- Simplicity
- Low channel and storage overhead
- Usage of up-to-date shortest routes
- Reliable construction of routes and forwarding group
- Robustness to host mobility
- Maintenance and exploitation of multiple redundant paths

- Exploitation of the broadcast nature of wireless environments
- Unicast routing capability

3.2 Terminology

3.2.1 General Terms

This section defines terminology used in ODMRP.

- node : A device that implements IP.
- neighbor : Nodes that are within the radio transmission range.
- forwarding group : A group of nodes participating in multicast packet forwarding.
- multicast mesh : The topology defined by the link connection between forwarding group members.
- join query : The special data packet sent by multicast sources to establish and update group memberships and routes.
- join reply : The table broadcasted by each multicast receiver and forwarding node to establish and update group membership and routes
- Member Table : Each multicast receiver stores the source information. The source ID and the time when the last Join Request is received from the source.
- Routing Table : It is created on demand and is maintained by each node. An entry is inserted or updated when a non duplicate Join Request is received. It actually provides the next hop information when transmitting Join Tables.
- Forwarding Group Table : A forwarding group member, maintains the group information in this table. The multicast group ID and the time the node was last refreshed is recorded.
- Message Cache : It is maintained by each node to detect duplicates.

3.3 Operation

3.3.1 Forwarding Group Setup

3.3.2 Originating a Join Query

When a multicast source has data packets to send but no route is known, it originates a "Join Query" packet. The Type field MUST be set to 01. TTL MAY be set to TIME_TO_LIVE_VALUE, but SHOULD be adjusted based on network size and network diameter. The Sequence Number MUST be large enough to prevent wraparound ambiguity, and the Hop Count is initially set to zero. The source puts its IP address in the Source IP Address and Last Hop IP Address field. It appends its location, speed, and direction into Join Query if nodes in the network are equipped with GPS.

When location and movement information is utilized, it sets the MIN_LET (Link Expiration Time) field to the MAX_LET_VALUE since the source does not have any previous hop node. When the source receives Join Replies from multicast receivers, it selects the minimum RET (Route Expiration Time) among all the Join Replies received. Then the source can build new routes by originating a Join Query before the minimum RET approaches (i.e., route breaks of ongoing data sessions are imminent).

3.3.3 Processing a Join Query

When a node receives a Join Query packet:

1. Check if it is a duplicate by comparing the (Source IP Address, Sequence Number) combination with the entries in the message cache. If a duplicate, then discard the packet. DONE.
2. If it is not a duplicate, insert an entry into the message cache with the information of the received packet (i.e., sequence number and source IP address) and insert/update the entry for routing table (i.e., backward learning).
3. If the node is a member of the multicast group, it originates a Join Reply packet with the RET value enclosed.
4. Increase the Hop Count field by 1 and decrease the TTL field by 1.
5. If the TTL field value is less than or equal to 0, then discard the packet. DONE.

6. If the TTL field value is greater than 0, then set the node's IP Address into Last Hop IP Address field and broadcast. DONE.

3.3.4 Originating a Join Reply

A multicast receiver transmits a "Join Reply" packet after selecting the multicast route. Each sender IP address and next hop IP address of a multicast group are contained in the Join Reply packet. The route expiration time is also included if the network hosts operate with GPS.

3.3.5 Processing a Join Reply

When a Join Reply is received:

1. The node looks up the Next Hop IP Address field of the received Join Reply entries. If no entries match the node's IP Address, do nothing. DONE.

2. If one or more entries coincide with the node's IP Address, set the FG_FLAG and build its own Join Reply. The next hop IP address can be obtained from the routing table.

3. Broadcast the Join Reply packet to the neighbor nodes. DONE.

3.4 Mesh Establishment in ODMRP

ODMRP uses the concept of forwarding group, which is a set of nodes responsible for forwarding multicast data on shortest delay paths between a sender and a receiver. An ODMRP source periodically updates routing tables and membership information by ooding the network with route refreshment packets, Join Query.The period of time between each Join Query transmission is called refreshment interval. Upon receiving a non-duplicate Join Query, an intermediate node stores the ID of the upstream node from which it receives the packet into the routing table, and then rebroadcasts the packet (duplicate Join Query packets will be discarded). When the Join Query packet reaches a multicast receiver, the receiver replies with a Join Reply packet, which contains the multicast source ID, and the corresponding next node ID from which it received the Join Query packet. The Join Reply packet is then relayed back towards the multicast source via the reverse path traversed by the Join Query packet.

When a node receives a Join Reply, it checks if the next node ID in the Join Reply packet matches its own ID. If it does, the node realizes that it is on the path to the source and thus is part of the forwarding group. It then sets a forwarding group ag and sends its own Join Reply further to a next node based on the routing table. The Join Reply is thus propagated by each forwarding group member until it reaches the multicast source. The route between a source and receiver is established after the source receives the Join Reply packet. This process constructs (or updates) routes from the sources to the receivers, and builds a mesh of forwarding nodes, the forwarding group.

An example of the mesh establishment phase is illustrated in Figure 3.1. For initializing the multicast mesh, sources S1 and S2 flood the network with Join Query packets. When

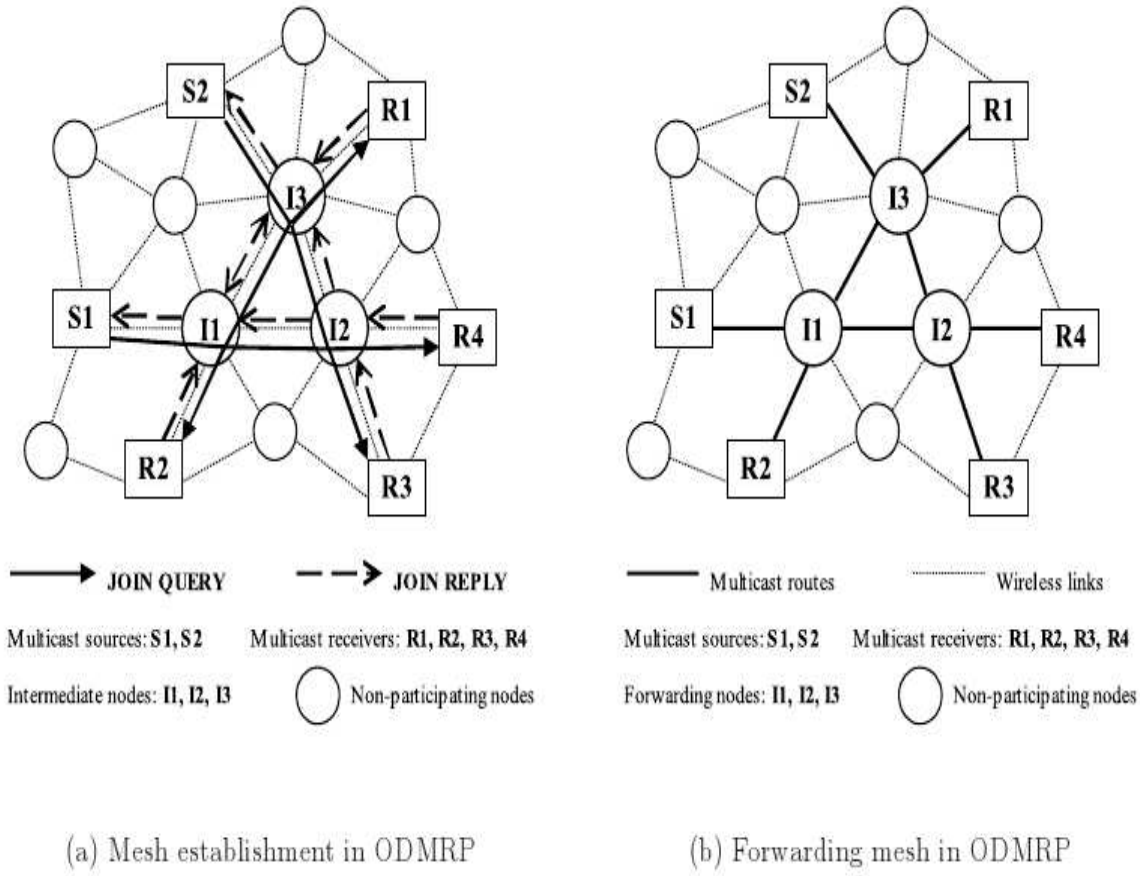

Figure 3.1: ODMRP multicast routing protocol

receivers R1, R2, R3 and R4 receive the Join Query packet, each node sends a Join Reply packet along the reverse path to the sources. For example in Figure 3.1(a), receiver R1 receives Join Query packets from sources S1 and S2 through paths S1-I1-I3-R1 and S2-I3-

R1, respectively. When node I3 receives the Join Reply packet from receiver R1, it sets the forwarding group ag and becomes the forwarding node for that particular multicast group. After sources S1 and S2 have received the Join Reply packets, a multicast mesh for sources S1 and S2 is established as shown in Figure 3.1(b).

3.5 Route Redundancy in ODMRP

Route redundancy in the multicast mesh helps ODMRP overcome frequent link breaks due to node mobility and channel fading in wireless communications. For example in Figure 3.1(b), suppose the route from source S1 to receiver R4 is S1-I1-I2-R4. If the link between nodes I1 and I2 breaks or fails, R4 can still receive data packets from S1 through an alternative route S1-I1-I3-I2-R4. This redundancy helps to achieve high connectivity among multicast members, and hence high percentage of packet delivery ratio.

Despite its useful application, multicast in MANET could not be practically deployed in real worlds without security consideration. The security problem in mobile ad-hoc networks faces many challenges. The next section summarizes previous security studies for mobile ad hoc networks.

Advantages of ODMRP :

1. Easy to implement
2. Highest packet delivery ratio
3. Robustness
4. Low overhead
5. Scalability to large number of nodes

Chapter 4

HEAP: Hop-by-hop Efficient Authentication Protocol

In this section, we describe the key steps in our proposed authentication protocol, HEAP. We then highlight the differences between HEAP, HMAC and NMAC in generating MAC (message authentication code).

4.1 Authentication for MANETs

Heap stands for Hop-by-Hop Efcient Authentication Protocol. Heap uses a modied HMAC-based Algorithm that utilizes two keys.

1. Shares a pairwise secret hash key, called okey, with each of its neighbors.
2. Generates one common secret hash key, called ikey, and securely distributes it to all of its neighbors.

4.2 HEAP Implementation

4.2.1 Key Generation and Distribution

- A Node that wants to join a network must first generate a single group key, called *ikey* (for inner key), and one pair wise key for each neighbor, called *okey*(for outer

key).

- Thus each node will generate *n+1* new keys (n *okeys* and 1 *ikey*), where n is the number of nodes in its neighborhood.
- The *ikey* is secretly shared with all the neighbors, while the pair wise *okey* is only shared with the corresponding neighbor. This can be done using standard key exchange mechanisms and trusted third party certicates, such as in the X.509 standard [10]. RSI or Elliptic Curve Cryptosystems (ECC) can be used as the underlying PKI.
- Whenever a nodes neighborhood changes due to mobility, it will share the *ikey* and a new okey with each new neighbor. TESLA, LHAP and Lu also have the same requirement where one or two keys are exchanged with each new neighbor as the node moves about.
- The keys should expire after a certain amount of time and new keys should be generated. This is to guard against brute force attacks and crypt-analysis attacks by the adversary.

4.2.2 MAC Generation and Authentication in HEAP

- After the key exchange phase, packet authentication can take place.
- Figure 4.1 shows, A is a node that needs to transmit a packet to its neighbors, A_1 through A_5.
- All the nodes have already been authenticated and node A secretly shares an *okey* with each of its neighbors, okA_1 to okA_5.
- In addition A has generated an *ikey*, *ik*, and has securely distributed to all of its ve neighbors. Only the one-hop neighbors of node A and node A itself have access to *ik*.
- Each recipient of the packet, A_1 through A_5, to be able to authenticate that the packet originated from node A.
- One way to do this would be for node A to generate a new HMAC for every individual neighbor using its okey. However, we present a more computationally efficient algorithm using a slightly modied HMAC. A key insight into the working of the HMAC algorithm [2] can signicantly lower the amount of computational overhead for multicast and broadcast applications.

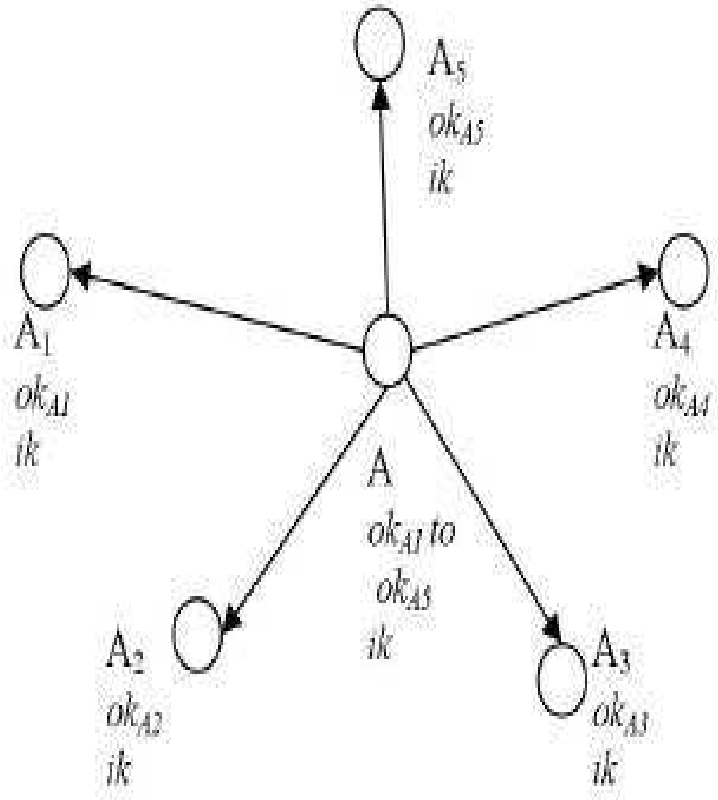

Figure 4.1: Node A needs to transmit a packet to its neighbors A_1 through A_5.

Recall that HMAC is computed in the following way:

$$HMAC(M, K) = H(K \oplus opad \mid H(K \oplus ipad \mid M)) \tag{4.1}$$

where H(x) is the hash value of x using some hash function such as MD5 or SHA1;
$\oplus$ is the XOR operation;
M is the message that needs to be sent;
opad is the hexadecimal number 5C that is used to pad each byte of K up to one block size, while ipad is the hexadecimal number 36 that is used to pad each byte of K up to one block size (i.e. 512 bits);
$\mid$ the symbol represents concatenation.

We divide HMAC computation into the following two steps:

1. Step 1: H(K $\oplus$ ipad $\mid$ M)

2. Step 2: H(K $\oplus$ opad | hash from Step 1)

For long messages,
Step 1 will result in most of the computational overhead.
Step 2 only needs to hash 32 bytes for MD5 and 40 bytes for SHA1. This insight led us to develop the following MAC generation algorithm for multicast and broadcast applications.

We use the previously described two keys, namely *ikey* and *okey*. We use *ikey* to generate the hash in Step 1. So the Step 1 for node A will be:

New Step 1: H(*ik* | M)

The ikey is padded with 0s to make it one block wide (i.e. 512 bits). Since all the one-hop neighbors share the same *ikey*, these neighbors can authenticate a packet received from node A by only computing this rst step. Therefore, the sender A only needs to compute this expensive step only once regardless of how many neighbors it has. However, we cannot use the same key for the second step as well. This is because all of As neighbors have the key and anyone of them could impersonate A and send forged packets. That is why we use the pair-wise *okey*, okA_i padded with 0s to make one block in Step 2:

New Step 2: H(okA_i | hash from Step 1)

Since only the sender and the receiver have okA_i no third party could generate this step. Of course, this step needs to be executed once for each neighbor. But fortunately, this step is computationally inexpensive because of the small input to the hash function of only 32 or 40 bytes.

Using the above MAC generation scheme, node A generates a packet using the following format and sends it to all its neighbors:

$A \rightarrow * : ind_A , M, MAC_1 (M \mid ind_A) \ldots MAC_n (M \mid ind_A)$

where * represents any recipient, M is the message, indA is the last index number used by node A, n is the number of neighbors, and MAC_i represents the MAC for neighbor A_i. A pseudocode for generating a packet in HEAP is shown below.

Packet generation in HEAP:
Input: Message M containing the payload and headers of upper layer,
index number ind_A , keys okA_1 to okA_n , and *ik*
Output: Packet S, new index number indA

---

1. **if** ind_A is null then
2. $ind_A := 1$
3. **else**
4. $ind_A := ind_A + 1$
5. **end if**
6. $mac_1 := H(ik \mid M \mid ind_A)$ // New Step 1
7. comp mac := {}
8. **for each** neighbor A_i **do**
9. $mac_2 := H(okA_i \mid mac_1)$ // New Step 2
10. comp_mac := comp_mac | last half of mac_2
11. **end for**
12. $S := ind_A \mid M \mid comp\_mac$
13. return S and ind_A

When A_i receives the packet, it computes the MAC and if it matches any of the MAC tags in the message and the index number is greater than the last authentic index number received the message is accepted as authentic; otherwise, it is dropped. To save on bandwidth the whole MAC need not be transmitted. Only the last half of the MAC can be transmitted in accordance with the recommendation made in [2]. This reduces the tag size and still makes the search space large enough for the adversary so that the chances of generating the correct MAC for a forged message are remote.

4.2.3 Index Numbers

Packet index numbers are included in the packet to protect against message replays. Before transmission, an index number is concatenated with the payload data to give the message M . Each time a packet is sent, its index number is incremented by one. The hash

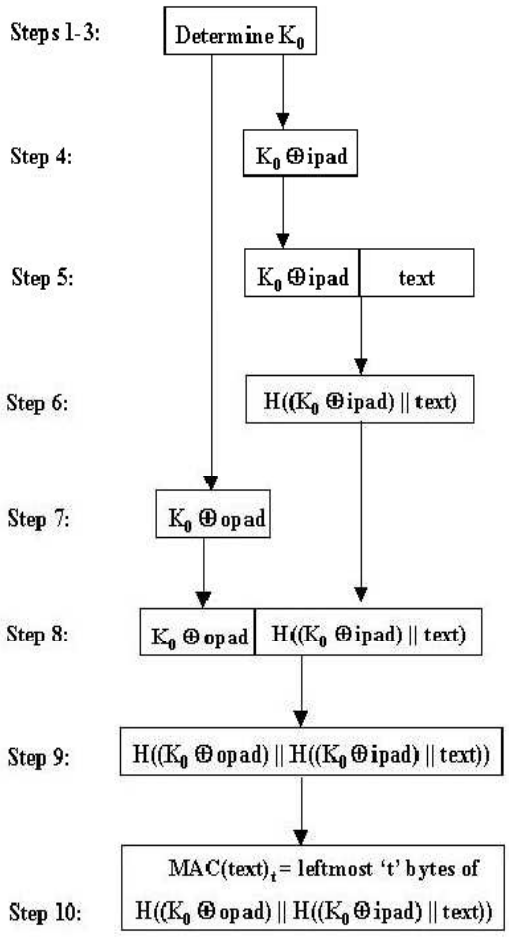

Figure 4.2: HMAC Construction

generated is dependent on the index number so any tampering of the index number can be detected. Any attempt to store and then retransmit the packet by a malicious node will be thwarted because the receiving node expects to see an index number greater than the last authentic index number and so it will consider this packet to be unauthentic.

4.2.4 Differences between NMAC, HMAC and HEAPs construct

Bellare et al. presented NMAC and HMAC in [2]. They listed two major dierences between NMAC and HMAC. First, NMAC (Fig. 4.3) uses two keys, k_1 and k_2 , while HMAC (Fig. 4.4) uses only one key, k. Second, NMAC replaces the xed Initialization Vector (IV) with the key. Normally, hashing algorithms such as MD5 and SHA1 use fixed IVs.

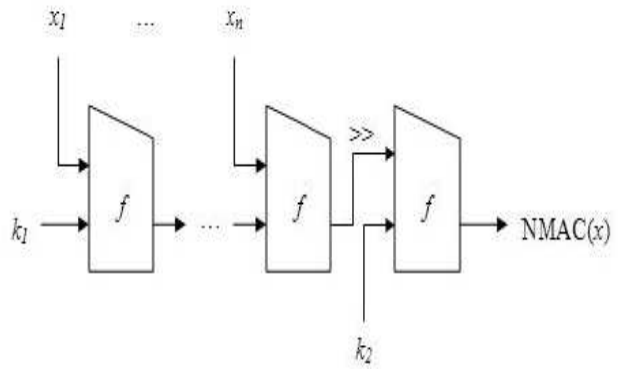

Figure 4.3: Illustration of the NMAC construct.

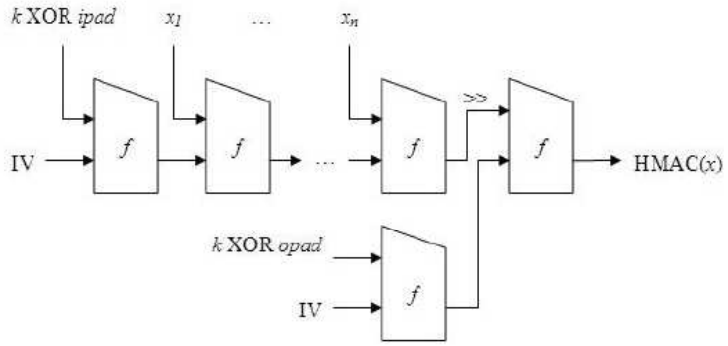

Figure 4.4: Illustration of the HMAC construct.

Replacing the IV with the key would necessitate a modication to the hashing algorithm so that it takes the IV as a parameter instead of keeping it xed. This would not be possible in o-the-shelf hardware implementations of MD5 or SHA1 and it would involve modifying any software library code for MD5 and SHA1. Although the change required in software would be trivial, it would still be preferable if no change was required at all and that was one of the motivations for the authors of HMAC.

In HEAP (Fig. 4.4), we wanted the ability to use Off-the-shelf hardware or software implementations of the hashing algorithm like HMAC, but we also wanted to use two keys instead of one like NMAC. Therefore, neither HMAC nor NMAC was completely suited to our purpose and we decided to modify the HMAC instead so it could accept two keys. In the figures above, the symbol >>represents padding by zeros to increase the size of the string from l bits to b bits, where l is 128 bits for MD5 and 160 bits for SHA-1, while b is 512 bits for both. The keys in HMAC and HEAP are also padded with zeros to make them

b bits long.

4.3 Security Analysis of HEAP

In this section, we present security analysis of HEAP and evaluate HEAP against our security goals that are stated in Section 2.4.

Evaluation against Goal 1: The first goal is to prevent all outsider traffic from propagating through the MANET. By denition, we assume that the outside attacker does not have any control over insider nodes, otherwise it is no longer an outsider but an insider. We can see from the HEAP algorithm that the only way an outsider can successfully propagate a packet is if it is able to generate the correct MAC for the packet. This means that the overall security of HEAP is dependent on the security of HEAPs cryptographic construct. Therefore, we now analyze the security of HEAPs cryptographic construct.

As discussed in Section 4.2.4, the primary difference between HMAC and HEAPs construct is that HMAC uses one key while HEAP uses two keys. However, as pointed out by its authors, HMAC actually uses two pseudorandom keys that are generated from one key. This is accomplished by padding the key with ipad and opad to generate two dierent seeds and then using the compression function on them to generate two pseudorandom keys. HEAP already has two different keys that can form the seeds for the compression function, eliminating the need to pad the keys. We still need to run the keys through the compression function, however. Otherwise we would need to replace the IV with the keys, like in NMAC, and that would prevent us from using o-the-shelf hash function implementations.

Because both HMAC and HEAP use two pseudorandom keys, we can apply the same security analysis for HMAC (or NMAC) to HEAP. For the case of an outsider that does not know any of the two keys in HEAP, the security analysis of HMAC holds and we can apply the same proof of security to HEAPs construct.

Evaluation against Goal 2: HEAP does not make any assumptions about the IP or MAC address of the packet. Even if these addresses are spoofed by an outsider, the outsider will not be able to propagate a packet unless it can generate the correct MAC for it. Therefore, HEAP is secure against IP and MAC address spoong.

Evaluation against Goal 3: In our analysis of HEAP, we made an even stronger assumption than simply that the attacker can eavesdrop on packets and view the message and its MAC tags. We considered the case where the attacker is able to choose its own messages and obtain their correct MAC tags, known as the chosen message attack. Therefore, our security analysis holds even if the attacker were able to view a very large number of messages.

Evaluation against Goal 4: As stated, HEAP is not designed to detect insider attacks. But if a third party Intrusion Detection System (IDS) were to detect a compromised insider node and alert other nodes about it, HEAP provides a framework for an eective Response System. The compromised node would be placed on a Certicate Revocation List (CRL) that would be propagated throughout the network. Nodes neighboring the compromised node would reinitiate the bootstrapping phase and exchange new keys with their legitimate neighbors but not with the compromised node. If the compromised node moved to another locality the other nodes would not exchange keys with it either because it is on the CRL. In this way, the compromised node would not be in possession of valid keys any more and it would be treated just like an outsider. We do not deal with the implementation details of the IDS and the CRL.

Chapter 5

Results and Conclusion

We presented a new Hop-by-Hop, Efficient Authentication Protocol, called HEAP. This protocol is suitable for use in MANETs for unicast, multicast or broadcast applications, and is independent of the routing protocol used. It is based on a modified HMAC algorithm that uses two keys and is very efficient. We compared the performance of HEAP with three other previously published authentication algorithms, namely TESLA, LHAP and Lu and Poochs algorithm.

TESLA is vulnerable to DoS attacks and requires secure time synchronization of all the nodes. It introduces very large latencies of several seconds making it unsuitable for real time or QoS applications. It also has huge memory requirements at every node and has mediocre throughput when the nodes are mobile.

LHAP is vulnerable to wormhole and man-in-the-middle attacks as pointed out by its authors. We do not think using GPS devices at every node, as recommended by its authors, is a practical solution. An attacker can successfully transmit forged packets using these attacks, defeating the purpose of authentication. It also has very large memory requirements at every node and consumes keys very rapidly. Once a node runs out of keys in its key chain, it needs to stop and regenerate the key chain causing cessation in its transmissions.

Lus scheme has poor overall performance. Their scheme significantly degrades throughput and packet delivery ratio, both in static as well as mobile nodes. In addition, it introduces large latencies making it unsuitable for real time applications.

HEAP is resistant to several outsider attacks such DoS, wormhole, replay, impersonation and man-in-the-middle attacks by making it very difficult for an outsider to propagate any forged packet. It has extremely low memory requirements and its CPU overhead is negligible, making it suitable for constrained wireless devices. Its byte overhead is also negligible since it has an insignificant effect on overall throughput and packet delivery ratio. Its latency is almost the same as without any security scheme, making it ideal for real time and QoS applications.

Bibliography

[1] W. S. Sung J. Lee and M. Gerla, "On-Demand Multicast Routing Protocol In Multihop Wireless Mobile Networks," *Mobile Networks and Applications, Kluwer Academic Publishers*, vol. 7, no. 6, pp. 441–453, 2002.

[2] R. C. M. Bellare and H. Krawczyk, "Keying Hash Functions For Message Authentication," *Proc. of Advances in Cryptology*, 1996.

[3] E. R. B. Dahill, B. Levine and S. C, "A Secure Routing Protocol For Ad-hoc Networks," *In Proceedings of the 10th Conference on Network Protocols (ICNP)*, 2002.

[4] P. Papadimitratos and Z. Haas, "Secure Routing For Mobile Ad-hoc Networks," *In Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2002.

[5] L. Zhou and Z. Haas, "Securing Ad Hoc Networks," *IEEE Network Magazine, special issue on networking security*, pp. 24–30, 1999.

[6] J. T. A. Perrig, R. Canetti and D. Song, "Efficient Authentication And Signing Of Multicast Streams Over Lossy Channels," in *In Proc. of IEEE Symposium on Security and Privacy*, 2000.

[7] S. S. S. Zhu, S. Xu and S. Jajodia, "Lhap: A Lightweight Hop-By-Hop Authentication Protocol For Ad-hoc Networks," *In Proc. of the 23rd ICDCS Workshop*, 2003.

[8] B. Lu and U. W. Pooch, "A Lightweight Authentication Protocol For Mobile Ad-hoc Networks," *International Journal of Information Technology*, vol. 11, no. 2, pp. 119–135, 2005.

[9] Y. Y. S.-J. L. W. Su and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks." RFC 2026, 2003.

[10] A. Aresenault and S. Turner, “Internet X.509 Public Key Infrastructure: PKIX Roadmap,” *IETF Internet draft, PKIX Working Group*, 2000.

[11] T. K. Rehan Akbani and G. Raju, “HEAP:Hop-by-Hop Efficient Authentication Protocol For Mobile Ad-hoc Networks,” *Elsevier Journal of Ad Hoc Networks*, vol. 6, pp. 1134–1150, 2007.

[12] Y. D. Luo Junhai, Xue Liu b, “Research On Multicast Routing Protocols For Mobile Ad-hoc Networks,” *Elsevier Journal of Ad Hoc Networks*, 2007.

[13] U. T. N. Hoang Lan Nguyen, “A Study Of Different Types Of Attacks On Multicast In Mobile Ad-hoc Networks,” *Elsevier Journal of Ad Hoc Networks*, vol. 6, no. 1, pp. 32–46, 2008.

[14] P. Ning and K. Sun, “How to Misuse Aodv: A Case Study of Insider Attacks Against Mobile Ad-hoc Routing Protocols,” *Elsevier Journal of Ad Hoc Networks*, pp. 795–819, 2005.