Georg-August-Universität
Göttingen
Institut Für Informatik

Anonymity and Privacy in Wireless Mobile Ad Hoc Networks

Dissertation
zur Erlangung des
mathematisch-naturwissenschaftlichen Doktorgrades
"Doctor rerum naturalium"
der Georg-August-Universität Göttingen

Vorgelegt von

Somayeh Taheri
Aus
Teheran-Iran

Göttingen 2011

Referent:
Prof. Dr. Dieter Hogrefe

Korreferent:
Prof. Dr. Jian Ren
Michigan State University

Mitglieder der Prüfungskommission:
Prof. Dr. Dieter Hogrefe
Prof. Dr. Xiaoming Fu
Prof. Dr. Jian Ren
Prof. Dr. Carsten Damm
Prof. Dr. Jens Grabowsky
Prof. Dr. Stephan Waack
Prof. Dr. Konrad Rieck

Tag der mündlichen Prüfung: 12 December 2011

Abstract:

Privacy has become a necessary property of networks as being considered as an important aspect of security by both network providers and customers. Privacy in general means to conceal one's personal information from others except the ones who are allowed to access those information. Nowadays, many different services and applications are subject to privacy protection, e.g. in every day Internet usage for instance when the user registers with a website, protecting his personal information is critical while he may feel having nothing to hide.

In ad hoc routing protocols, privacy becomes an important issue when mobile ad hoc networks enter security critical domains. In ad hoc networks, due to the lack of a centralized support, every device relying on other nodes for data transmission is responsible for its own routing operations and security issues. In this dissertation we present the methodology, design and the results of our research to achieve further steps toward private communication in ad hoc networks in hostile environments.

Privacy in wireless environments is divided into two types: identity anonymity and location privacy. Identity anonymity, as the main anonymity property, has attracted a lot of attention among researchers during the last decade. Identity anonymity has been addressed in many MANET (Mobile Ad hoc NETwork) routing protocols while location privacy is attracting an increasing attention and still needs appropriate solutions. Almost all of the existing works are designed to provide identity anonymity for MANETs rather than location privacy. In the routing protocols that provide identity anonymity, a strong enough traffic analyser could still track the nodes' locations in active routes [Y.C. Hu 2005].

Privacy in ad hoc networks is a challenging issue due to the common vulnerabilities of wireless mobile networks. The attacker can launch traffic analysis against the routing information as well as eavesdropping the packets and tracing message flows in the network to discover their origin using an appropriate set of directional antennas. We believe that one of the important existing problems in the area of ad hoc network security is location privacy.

Besides, although operating as groups is required by many ad hoc applications there is only a sparse work on anonymous multicast routing algorithms in the literature. Regarding the lack of appropriate anonymous multicast routing protocols for MANETS, privacy in one-to-many communication is a challenge remaining to be solved in this kind of wireless networks.

This thesis investigates solutions for network elements' anonymity and location privacy in unicast and multicast applications of MANETs. We propose a mechanism for destination location privacy in unicast ad hoc routing against a global eavesdropper as well as a framework for anonymous multicast routing for

mobile ad hoc networks. We will provide both privacy analysis and implementation to evaluate the performance of our ideas. The evaluations demonstrate that the proposed techniques are successful in achieving the privacy goals while keeping the performance in a high level.

Keywords: privacy, anonymity, location privacy, ad hoc networks, MANET, pseudonymity, security, multicast, group communication, mesh.

Acknowledgements

First of all, I would like to express my deepest gratitude to my advisor Professor Dr. Dieter Hogrefe for his supervision, advice and guidance during my PhD. work while allowing me to work in my own way. I am grateful to him because of his great support and encouragement both at the professional and personal levels.

My respect and gratitude also goes to Prof. Xiaoming Fu as my secondary advisor. I owe many thanks to Prof. Jian Ren for reviewing my dissertation and also for giving me the opportunity of having a visit at his group in Department of Electrical and Computer Engineering at Michigan State University.

I am specially thankful to Salke Hartung for his scientific collaboration and for the helpful discussions that we had during my research.

I would like to thank all the members of Telematics group in Georg-August University of Göttingen, specially my good friend Parisa for the nice times that we had together. Having the friendly and prompting working environment in Telematics group was not possible without the professional efforts and kind supports of Carmen Scherbaum and Udo Burghardt, I thank them very much.

My husband deserves a big thank for supporting me all the time patiently and kindly in various ways. Words fail me to express my appreciation to him for his dedication and love and for his support by sharing his valuable experiences.

My parents receive my deep sincere gratitude and love for their support and encouragement over many years. I dedicate this modest work to them.

Göttingen, October 2011 Somayeh Taheri

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ad hoc networks, as self-organized collection of mobile users, need different protocols than the ones proposed either for wired networks or centralized wireless systems. Ad hoc networks, due to the special requirements of their applications, are designed such that no central devices such as routers, intrusion detection systems or central trust management would be a part of the network. In these infrastructure-less networks, each node as a sender needs to find its route to the desired destination itself through the potentially unknown intermediate nodes and it depends on the network protocols how much risk is taken regarding his personal information privacy, the data integrity or confidentiality.

In critical applications of ad hoc networks the presence of active or passive adversaries, who may try to discover some private information of the network or destroy the network operations, is always likely. In addition, due to the limited power or data storage resources in ad hoc nodes, some of internal network members also might act selfish in routing functionalities. In such situations, the protocol design plays a very important role in keeping the network safe and private to achieve the application's goals. For example regarding information privacy the protocol could be designed such that even if a node locating on a communication path is compromised by the adversary and therefore the information stored on it are disclosed to him, the adversary still cannot access the source/destination nodes' identification information.

In the remainder of this chapter, we will describe the objectives of this dissertation and will define the important terms used in this dissertation related to privacy issues as well as giving a description of ad hoc networks, their properties and security issues.

1.1 Objective and scope

The main objective of this thesis is to provide the layout of efficient private communication in ad hoc networks and is divided into two goals. The goals followed are to develop solutions to address part of privacy issues in mobile ad hoc networks, in both unicast and multicast scenarios. More specifically, we will propose a solution to protect location privacy of destination node in a unicast communication and also will design an anonymous multicast routing protocol for MANETs providing location privacy for the key elements of a group based communication.

The scope of this dissertation includes anonymity and location privacy in one-to-one and ono-to-many communications in mobile ad hoc environments.

1.2 Terminology and background

1.2.1 Privacy

Privacy as a word comes from the Latin word "*privatus*" which meant "separated from the rest, deprived of something" [wik]. Data privacy aims to transfer or share data with the desired parties while the personally identifiable data is concealed from others who may even attempt to access that. There are various definitions for privacy which look different but share a common base. Here we refer to the one by Alan Westin [Westin 1967].

Definition 1.2.1. ***Privacy*** *is the claim of individuals, groups and institutions to determine for themselves, when, how and to what extent information about them is communicated to others.*

In wireless network protocols, privacy is typically about the entities' identities, locations or relationships. For example, one ad hoc network military application may concern the ID privacy of the source node and a sensor network object detection application may concern the location privacy of the reporter node.

There is not a global definition for network privacy and actually it depends on the purpose of the network which determines what properties or contents must be kept private. We express our definition for network privacy in mobile wireless networks as follows.

Definition 1.2.2. ***Network privacy*** *is the state of making it impossible or difficult for third parties to obtain confidential information of any network entities or current or permanent properties disclosed in a private place to trusted parties.*

1.2.2 Anonymity

Anonymity is derived from the Greek word, anonymia, meaning "without a name" or "namelessness". Anonymity basically means the claim of being publicly unidentifiable. For its definition we refer to the definition by Pfitzmann and Hansen [Pfitzmann 2008].

Definition 1.2.3. ***Anonymity*** *of a subject means that the subject is not identifiable within a set of subjects, the anonymity set.*

This means that the anonymous entity can do his job without being identified by the attacker while the attacker might be able to recognize a set of subjects which he knows includes the subject of interest. Anonymity is not an absolute

concept, i.e. its existance or amount depends on the ability and the knowledge of the attacker. One subject could be anonymous to one guy while he is identifiable to another one. For example, when all of the students are sitting in the classroom their teacher can recognize Alice among the others but Bob's father cannot since he does not know Alice by face, then Alice is anonymous to Bob's father but not to her teacher.

Also anonymity could range from very weak to very strong in continuous levels. For example, assume that Bob's father is told that Alice is in the classroom. In the first scenario he sees 9 girls and Bob in the classroom, so to him any of the 9 girls could be Alice with the probability of $\frac{1}{9}$. In the second scenario he sees 5 girls and 5 boys in the classroom, so to him any of the 5 girls could be Alice with the probability of $\frac{1}{5}$. Obviously, in the first scenario Alice's anonymity degree is higher.

There are many reasons of wishing to be anonymous. One way of being anonymous is to use false or even stolen identities and another way is to use pseudonyms.

1.2.2.1 Privacy vs. anonymity

As the definition of anonymity indicates, anonymity of a subject is mostly about its identity. As mentioned before privacy means to protect one's personal information from being disclosed to the outsiders. Personal information could be of different types, e.g. one's identification, user name, student number, place of residence, etc. Therefore the relationship between anonymity and privacy could be clarified as follows:

- *Anonymity is one aspect of privacy*, or: anonymity is identity privacy.

1.2.3 Pseudonymity

"Pseudonym" comes from the Greek word "pseudonumon" meaning "falsely named" (pseudo: false, onuma: name). Sometimes the object who is willing to be anonymous chooses to use a pseudonym instead of its real identity. A pseudonym could be any attribute of the user rather than his real identity used for identification in a temporary or permanent relationship such as a randomly generated identifier or a student number. The pseudonym helps the other party to link different messages from the same individual. Consequently pseudonymity means the state that one introduces himself using a pseudonym in a network. We use the definition of pseudonymity given in as follows [Pfitzmann 2008].

Definition 1.2.4. ***Pseudonymity*** *is the use of pseudonyms as IDs.*

Pseudonymity itself does not tell anything about the strengths of anonymity, authentication or accountability [Pfitzmann 2008]. Pseudonymity aims to pro-

vide anonymity while the pseudonymous object is successfully authenticated and accountable to the set of users.

1.2.3.1 Anonymity vs. pseudonymity

According to the above definition, although pseudonymity can be considered as another concept but it can be seen as being anonymous by using identities rather than the real one. Therefore it is argued that:

- *Pseudonymity can be seen as a kind of anonymity.*

1.2.4 Mobile ad hoc networks

The word *ad hoc*, translated as "for this purpose" from Latin, is used for the spontaneously constructed networks to deal with immediate demands such as a short term local communication. The difference between this kind of networks and other existing networks is that they do not use any fixed infrastructure, i.e. they have no base stations, access points, remote servers, etc. The nodes in ad hoc networks do not have access to high power supplies and are restricted to their battery power capacity which could not be charged so often. So energy efficiency in ad hoc networks is an important required property of ad hoc protocols. Ad hoc nodes function as hosts as well as routers in the network to route data packets between communicating pairs. Mobile ad hoc networks are autonomous networks interconnected in a multi-hop manner via wireless links between the nodes [Merwe 2007]. To have successful and efficient ad hoc networking, cooperation between nodes is more than necessary. If some nodes act selfish in packet forwarding for others, the rest of the nodes will have problems or failures in their connections or will suffer from a high data forwarding overload.

Game theory can be used as a tool to analyse ad hoc nodes' behaviour in different scenarios as it is invented to model multi-member decision making systems in which the members may wish to maximize their utility.

Application areas

MANETs as self-configuring networks of mobile devices are suitable for use in areas where rapid network reconfiguration is required. They are mainly used where an infrastructure is not available or it is not efficient or cost effective to develop one. One common property of ad hoc applications in that they are all localized.

- Bluetooth

  The most famous application of ad hoc networks is Bluetooth, which is designed to support a personal area network to exchange data in short distance between devices such as mobile phones or PCs and printers.

- Military applications

  Military applications are also one important category of ad hoc networking applications. Armed forces involved in offensive or peace keeping missions need to exchange information by communicating voice or data about the current battle field situation while they are moving in the field. As for ad hoc networking, it is enough that every node is in the radio range of one neighbor, ad hoc networks are the best choice for such applications.

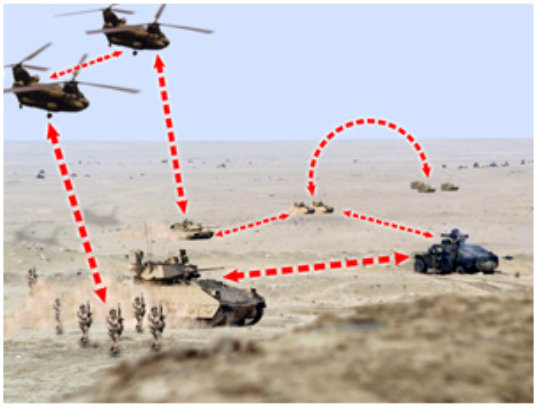

Figure 1.1: Ad hoc networks application: military communication

- Temporary networking

  Ad hoc networks can also be used in meetings, conferences and conventions where participants need to share information dynamically using their mobile devices. Even, for example, when a group of travelling scientists in an airport wish to share some data they can form an ad hoc network by switching the radio network interface cards of their laptops to the ad hoc mode.

  Disasters are another situation in which ad hoc networks are suitable to use. For example shortly after big earthquakes, as the local infrastructures might be destroyed and not usable any more, ad hoc networks can be used for search and rescue operation.

- Vehicular networks

  Another important area that ad hoc networks are used is VANETs (Vehicular Ad hoc NETworks). In VANETs, ad hoc devices are installed in automobiles to facilitate local communication between several cars or between a car and a roadside access point [Xiaodong Lin 2008]. In vehicular ad hoc networks, two cars that are far away from each other in the network can exchange data using intermediate cars while everybody is moving.

- Sensor networks: e.g. smart buildings and environmental measurements

  Ad hoc nodes designed as sensors are suitable for many applications such as habitat monitoring, environmental parameters measurement applications or biological detections. Sensors also can be deployed in smart houses to create a sentient computing environment in the building [Cayirci 2009].

1.2.5 Security and privacy vulnerabilities of MANETs

The special properties of ad hoc networks as wireless infrastructure-less networks makes security goals important and also difficult to achieve in them. Moreover, although encryption of packets can help to defeat some security threats such as information integrity, it would not be enough where privacy becomes important to the network application. For example in some military missions or police operations it may be necessary to hide who is communicating to whom. To avoid potential privacy threats one would like to achieve identity anonymity or location privacy. Location privacy means to hide the current or past venue of a network entity from the outsiders. An obvious example of need to ID anonymity is the concern of the source node about disclosure of his ID embedded in the routing packets which might be overheard by the eavesdroppers sitting in the radio range of the intermediate nodes, and an example of need to location privacy is the concern of a group-based network about the disclosure of the location of the group leader who is responsible for group membership management. If the privacy of the nodes is not protected successfully, denial of service attacks could be lunched by the adversary against the nodes when they are identified by him.

In this section we briefly describe how the characteristics of ad hoc networks lead to their general security and privacy vulnerabilities.

- *The insecure and broadcast nature of wireless medium:*

  ◦ Active attacks $\Longrightarrow$ Security issues

  Use of wireless links make ad hoc networks susceptible to active link attacks. For example the adversary may lunch message replay, message distortion and denial of service attacks against the network exploiting the possibility of accessing the wireless medium by either receiving (listening to) the ongoing communication or inserting its own packets in the links to achieve its goals.

◦ Passive attacks $\Longrightarrow$ Privacy issues

In passive eavesdropping attacks the data transmissions can be overheard by anyone close enough to the active routes. Passive attacks in wireless networks are very closely related to privacy, because every network member's activities could be detected and traced by the eavesdroppers who are monitoring the network to see nodes' radio transmissions using directional antennas. A passive attacker performs no activity in the network usually because he wants to be undetectable.

- *Poor physical protection in hostile environments:* As mobile ad hoc nodes are roaming in a hostile environment with relatively poor physical protection, they are very likely to be compromised by the adversary. We consider a captured ad hoc node as an internal adversary as it will be under the control of the adversary upon being compromised, i.e. not only a compromised node is lost from the network, but it potentially can act as an adversary against the network goals.

- *Dynamic topology and membership:* In mobile ad hoc networks the network topology is very dynamic since the nodes can move arbitrarily in the network area. A node who is part of a route may move at any time such that the route gets disconnected. Trust relationships between neighboring nodes also changes when nodes are moving in the area which means selfish or hostile behaviour of nodes cannot be monitored easily to achieve secure communication.

- *Lack of a central trusted authority:* Since ad hoc networks are usually deployed without use of any infrastructure in the environment, they cannot have any central supports. In addition the node capture possibility actually indicates that it is not a good idea to have a central authorization in the network. Including any central entity in the network means that the entire network might fail if the centralized entity is compromised. So ad hoc networks are deployed as distributed architecture which means that security and privacy management has to be distributed in the network.

- *Malicious nodes:* Since there is no router in ad hoc networks and the ad hoc nodes act as routers simultaneously, the network members have to rely on each other to transmit their data packets. In hostile environments there might be malicious nodes in the network who aim to deliberately disrupt or deny the normal routing operation by discarding or modifying sensitive routing information. For example, a malicious node can modify the route length metric to mislead the source node to find the shortest path to the destination. Trust management deals with preventing malicious nodes from participating in routing operations. Trust management in ad hoc networks as dynamic networks is more difficult compared to fixed and static

networks. In MANETs trust management is a dynamic system problem and due to the limited resources it should only rely on local information which leads to some limitations.

- *Limited resources:* Ad hoc nodes depend on their limited battery power for their entire activities in the network. Also their data storage capacity is usually restricted. Therefore, every packet forwarding or routing information storage for other nodes means consumption of its limited resources to a node. Security and privacy protocols mostly require more attempt by the users to protect the network security and privacy in terms of key agreements or cryptography operations. Therefore, achieving security is much more challenging in MANETs.

Any security solution designed for traditional or static configurations would not address ad hoc networks' issues directly. The same argument applies to network privacy. Anonymity mechanisms available for wired networks are not suitable for ad hoc networks because they depend on a centralized support which cannot be available constantly in ad hoc network. They also may rely on using some appropriate dummy packet transmissions to hide the real transmission when they must be concealed due to privacy requirements. This property does not fit MANETs also, because of the power and bandwidth limitation of ad hoc network.

1.2.6 Location privacy issue in ad hoc networks

In security critical domains both identity and location privacy could become very important and valuable properties required by the applications. Even in routing protocols supporting ID anonymity, a strong enough traffic analyser may still track the nodes' locations in active routes [Y.C. Hu 2005]. The goal in location privacy is to protect the information about the nodes' venue specially the end nodes. As soon as the adversary achieves his favourite information about the network members' venues, he will be potentially ready to attack the network functionalities by performing different kinds of denial-of-service attacks. The adversary could achieve such information by tracing some message flows in the network to discover their origin using an appropriate set of directional antennas.

1.2.7 Multicast communication and security challenges

Many of ad hoc applications require multicast communication which means collaboration between network members to work as a team. For example, when network members need to share data or files in a meeting or when a single user needs to show audio/video/images to others as in audio/video conferences, or when a commander needs to send a command to several soldiers in a military

mission. Multicast in wired and wireless networks aims group orientated, i.e. one-to-many or many-to-many, communication in a way that all destinations are identified by a single destination address i.e. the multicast address. Among the benefits of multicast networking is a reduced network overhead and bandwidth consumption in both wired and wireless mediums. In wired networks the data packet will be copied at the intermediate routers on the way to reach multiple receivers till it is delivered to all of them and in wireless networks multiple copies of the message are transmitted over the multicast structure by exploiting the inherent broadcast nature of wireless transmission. This in turn leads to scalability of the network which is another benefit of multicasting. Regarding privacy, multicast receivers can receive data while their individual addresses are unknown or changeable to the sender [Kunz 2004].

Multicast security aims to deploy multicast systems with proper key management, data confidentiality, integrity, access control, user authentication, Non-repudiation having possibly precautions against denial of service attacks. Achieving security and privacy in multicast is even harder than in unicast. In multicast systems the connections between the sender and each receiver cannot be controlled easily since it can go through a huge and complicated mesh. Issues such as group dynamic membership, the structure of connections between group members, group size, multiple group networking, group connection repairing, group-based routing algorithm are required to be managed in any multicast scenarios.

Multicast key management is more complicated compared to unicast and loads much more traffic to the network to distribute the keys. Usually the unique group key should be shared among the whole group including group senders and receivers. For message authentication the group senders must distribute their authentication keys to the group receivers. In some multicast protocols one core node responsible for group key distribution is introduced to perform the key management operations for the members at the time of their join to the multicast system.

In multicast communication the sender needs to deal with data privacy over not only one established route to one receiver but the connections to a group of receivers. If there is a central entity to lead the group for any purpose, protecting its identity and specially location privacy would be of a high level of importance as the whole group communication security could depend on it. We believe that multicast anonymity is one of the important current challenges of ad hoc networks, while among the proposed anonymous algorithms for ad hoc networks there is only very limited research on group based applications.

1.3 Problem statement

In this thesis we concentrate on current issues of anonymous and private communication in mobile ad hoc networks, with a special focus on location privacy. The research questions that will be answered in this dissertation are:

1) Considering security critical applications it can be of a high importance to physically protect the receiver of the communication. How it could be possible to prevent the attackers from finding the venue of the destination node? How such a goal can be achieved if the adversary is a global eavesdropper who is monitoring the whole network to detect any signalling activities?

2) Regarding the importance of group-based communication for many applications of MANETs specially in hostile environments, how to develop anonymous multicast communication in which the identity and location privacy of key group members is protected?

1.4 Thesis contributions and organization

First we will discuss security and privacy issues of ad hoc networks and the existing solutions proposed to deal with them in Chapter 2.

Then in chapter 3 we will present an approach to achieve destination location privacy in unicast communication against a global eavesdropper. The existing anonymous ad hoc routing protocols typically deal with the identity privacy of the end nodes but not their location privacy. In this approach we propose a solution to hide the venue of the destination of the communication from the outsiders using a message type unification mechanism. This approach also provides route privacy to prevent the adversary from finding the end points by following the route and also from being able to do link attacks against the route itself. A mathematical privacy analysis is provided to measure the level of privacy of the protocol and the performance of the protocol is evaluated by implementation.

Chapter 4 introduces a framework for anonymous multicast routing protocol for MANETs providing location privacy protection for important nodes. This framework rests on the following contributions to provide private group-based communication.

This approach develops a receiver initiated mesh based multicast routing protocol concerning anonymity of the nodes as well as location privacy of the group senders and the leader. The idea of identity free communication proposed in [Kong 2003] is extended to multicast routing in mobile ad hoc networks and a message type unification mechanism is used to hide the group activities inside the mesh. Both privacy analysis and performance evaluation results are presented.

The main design challenges in both protocols have been as follows. First, the solution should be distributed among the system elements due to the distributed nature of ad hoc networks. Second, it is supposed that a global eavesdropper is monitoring the whole network as the adversary who is potentially able to trace any activities of the nodes or the group by analysing the whole network traffic.

After describing and analysing our work, a conclusion of this research will be presented in chapter 5.

Chapter 2

Security and Privacy in Ad Hoc Networks

2.1 Security and Privacy Threats

In this section we present taxonomy for attackers and the attacks carried out by them in both security and privacy contexts. We classify attacks based on the OSI layers that they threaten. Existing countermeasure methods used by networks in hostile environments and some security and privacy enhancing routing protocols are also described.

2.1.1 Attacker classifications

As the hostile entities in a network may have different purposes, their adversarial activities can be of different classes. They range from passively observing the network flow to actively disrupting the network operations, and according to their domain they can be insider or outsider adversaries. Therefore, as mentioned in many articles the attackers in ad hoc networks are classified as follows.

- Active vs. passive attackers

  An active attacker disrupts the network functionalities by dropping, replaying or changing messages, as well as impersonation, preventing hosts from accessing normal network services, disrupting the routing operations, etc.

  Passive attackers are those who try to hide their presence from the network and wish to be undetected. A passive attacker does not interrupt any network functionalities like an active adversary but he is still an attacker by obtaining information about what is going on in the network. He may look for information about the roles of the nodes e.g. who is communicating to whom. He eavesdrops the packets, monitors the network flow and analyses the network traffic to achieve his favourite information about the network status threatening network privacy. It can be argued that passive attacks include eavesdropping, traffic analysis, and traffic monitoring [Mamatha 2010]. It would be very difficult to identify such attacks since

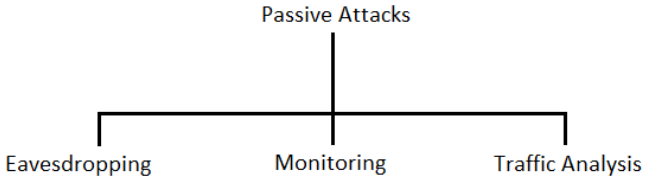

Figure 2.1: Passive attacks

network itself is not affected. Passive attacks against the on-the-fly data can be defeated to some extent by using powerful encryption techniques [Mamatha 2010]. Some other passive attacks attempt to break the location privacy of important nodes in the network, e.g. active sender/receivers, by tracing them by traffic monitoring using a set of directional antennas.

- Internal vs. external attackers

  An internal attacker is a network member or is able to make control on a network member (compromising him). Therefore, an internal attacker is possessing a valid network ID and access right to the network and as a part of the network knows valuable secrets of the network.

  External attackers do not belong to the network and attempt to damage the network services without knowing the network secrets or having any access right initially. Obviously it is easier to deal with external adversary rather than insiders.

2.1.2 Security attacks: A layer-based overview

Security attacks can be performed against different layers of the protocol stack. In the following we will present a layer-based classification and description of adversarial attacks in ad hoc networks. Table 2.1 presents the classification of attacks in MANETs based on OSI model layers.

- **MAC layer attacks**

  MAC layer attacks are lunched against the availability of the Mac layer, e.g. attacks against power saving mechanisms or RF jamming attacks.

Table 2.1: An attack classification in MANETs

MAC layer attacks	RF jamming and interface attacks MAC layer flooding attacks
Network layer attacks	*route discovery phase attacks:* routing table overflow blackhole attack, wormhole attacks, rushing attacks neighbor attacks and colluding misrelay attack *Data forwarding phase attacks:* message modification attacks message replay attacks and link-brocken message attacks grayhole attacks, flooding attacks, jellyfish attacks
Transport layer attacks	session hijacking attacks, SYN flooding attacks
Application layer attacks	repudiation attacks
Multi-layer attacks	impersonation attacks, sybil attacks, DOS attacks man-in-the-middle attacks

  - *RF jamming and interference attacks:* Jamming attacks are actually kinds of denial of service attacks to MAC layer (and physical layer) in wireless networks. Since the wireless medium is open to everybody including the present adversaries, the jammers would be able to deny services to legal network users by lunching various kinds of interference at the frequency channels. There are different types of jamming attacks regarding the way of frequency interference, e.g. constant jamming, deceptive jamming and random jamming. It is not always easy to realize that a jamming attack is happening, but the fade in signal to noise ratio can indicate the presence of a jammer if no better reason if found for it.

  - *MAC layer flooding attacks:* In MAC layer flooding attacks the attacker transmits packets using a spoofed source MAC address. One example of such MAC layer attacks is authentication/association flood attack, in which the attacker uses a spoofed MAC address and repeatedly makes authentication/association requests till the memory and the power of the target nodes is exhausted [Compton].

- **Network layer attacks**

  Various kinds of attacks against network layer aim to disrupt the normal routing operation in order to damage the network communication. Some of them happen at the route discovery phase and the others are lunched

after the route is established.

– *Route discovery phase attacks:*

- *Routing table overflow:* In this kind of attack the adversary attempts to fill the routing tables of nodes by establishing routes to non-existing destinations. For example the attacker may reply to any route request message advertising having a route to the intended receiver, or two or more hostile nodes may collude to establish routes by replying to route requests generated by themselves. Consequently, the normal routing operation of the network may fail after the routing tables are occupied by such forged and useless entries.
- *Blackhole attacks:* In the routing process where the source node is finding the shortest path toward the destination node, the attacker advertises himself as having the shortest path to the destination. As the most protocols are interested in shortest path in routing process the malicious node would locate on the discovered route. Therefore he will have the possibility of dropping data packets or performing message modification attacks. The attacker is called the Blackhole on the route.
- *Wormhole attacks:* The adversary establishes a connection between two different adversarial nodes, let's say A and B, in two different areas of the network. Then node A forwards everything he hears in his own neighborhood to node B and node B replays them. the same happens in the opposite direction. Consequently, the nodes near node A suppose themselves as neighbors to the nodes in area of node B and vice versa. As a result, the route discovery operations taking place after that can be affected by the wormhole link since the A-B shortcut could be a part of them and the data packets will be discarded by A or B. The wormhole link between A and B may be established by using an ethernet cable, an optical link, etc.
- *Rushing attack:* The goal of this attack is to invade into paths between senders and receivers. In almost all ad hoc routing protocols each intermediate node processes only the first route request message received from the source and discards the duplicate packets arriving later on. In rushing attack the attacker rushes the route request packets by forwarding the RREQ packet quickly skipping processing or routing steps. The result is that the source node cannot find any reliable route to the destination since the attacker is located on the discovered route.

- *Neighbor attacks:* The attacker forwards the routing packet without recording its participation in the routing process in that. It means that its two neighboring nodes believe to be neighbors while the attacker is locating between them on the route. Consequently, the data forwarding will fail unexpectedly.

- *Colluding misrelay attacks:* Multiple colluding adversarial nodes disrupt route discovery operations by dropping or modifying routing packets. For example consider the following scenario. Nodes A_1 and A_2 are neighbors and are colluding to drop routing packets knowing that the network uses watchdog mechanism to prevent this. Node A_1 forwards the routing packets received from its upstream node N_1 to prevent being detected as a malicious node by N_1. But node A_2 drops the packet. So, the watchdog mechanism of the network will not detect the packet drop because node A_1 as the colluding node will not report the packet drop misbehaviour. Therefore, the colluding misrelay attacks are not easy to detect [Pradip M. Jawandhiya 2010].

– *Data forwarding phase attacks:*

- *Message modification attacks:* In message modification attacks the attacker intercepts one or more messages and then modifies and retransmits them. Any activity against data integrity is considered as a type of message modification. Message reordering is also one type of message modification attacks [Martucci 2009]. In wireless networks, due to the broadcast nature of the wireless links it is easy for the active adversaries to lunch message modification attacks.

- *Message replay attacks:* In replay attacks the attacker records a message from the valid transmissions between two parties and then maliciously retransmits or delays that in the network. An attacker can lunch a replay attack to authenticate himself to a network party by retransmitting a recorded password. Replay attacks are some times considered as Man-in-the-Middle attack. Time stamping or session tokens are solutions to prevent this kind of attacks.

- *Link-broken error message attacks:* The attacker intrudes into the path between source and destination nodes. Then he initiates an error message indicating that his link on the route is broken (which normally could happen due to node movements) which causes the routing protocol to repair the route or discover a new route between the communicating pair.

- *Grayhole attacks:* Grayhole attacks are weaker than blackhole ones and can be considered as a variant of them. In a grayhole attack the adversary first behaves normally in routing process but will drop some or all data packets in data forwarding phase. Honest nodes might drop packets due to network congestion, so detecting a grayhole attack is not easy.

- *Flooding attacks:* The nodes' resources such as battery power, memory or bandwidth are consumed by the flooding attacker to forward his large amount of traffic (usually routing packets). For example the attacker may broadcast RREQ packets intended to destination nodes addresses that never exist in the network. Every node in the network will have to forward such routing packets for no benefit to the network. There have been some solutions proposed to reduce the possibility of flooding attacks. For example, the number of RREQ that can be originated per second is limited [Ping Yi 2011]. Flooding attack is considered as a kind of denial of service attacks.

- *Jellyfish attacks:* The attacker needs to intrude into the route between the source and the destination. Then he delays the data packets on purpose before forwarding them. Therefore the end-to-end delay and also the delay jitter will increase [Nguyen 2008].

- **Transport layer attacks**

  – *Session hijacking attacks:* In session hijacking attack the attacker disrupts the ongoing session usually by denying the victim entity from the current service. For example in a TCP session the attacker after finding out about the current expected sequence number of the data packets spoofs the entity's IP address and impersonates him to the other party for the rest of the session [Pradip M. Jawandhiya 2010].

  – *SYN Flooding attacks:* The goal of TCP-like Transport layer protocols in ad hoc networks is to create reliable end-to-end connections flow control and control the flow congestion in them. Such connections could be subject to SYN flooding attack which aim to exhaust the resource of the victim entity [Y. Xiao 2006]. The attacker sends many TCP connection requests to a host using a spoofed source addresses but never completes the handshake to fully open the connection. It will cause the attacked host not to be able to establish the next legitimate incoming TCP connections leading to a denial of service situation.

- **Application layer attacks**

  - *Repudiation attacks:*
    Repudiation attack happens when a malicious user is accessing the network while denying completely or partly of participation in the network communications.

- **Multi-layer attacks**

  Multi-layer attacks are those who can occur in all or several layers of the protocol stack.

  - *Impersonation attacks:* Impersonation attacks happen when a node pretends to be the owner of another node's ID. If the attacker impersonates two or more other entities the attack would be called Sybil attack as will follow. Message sequences can be replayed and data link addresses can easily be spoofed in wireless networks [Michel Barbeau 2006]. A strong authentication mechanism can prevent such attacks. As impersonation attacks are lunched before many disruptions to different network functionalities, this type of attack is considered as a multi-layer attack.

  - *Sybil attacks:* As mentioned Sybil attacks are a kind of impersonation attacks. In Sybil attacks the hostile node appears as the owner of several node IDs. The multiple identities of the attacker can be fabricated or stolen IDs which are known as Sybil IDs. The amount of Sybil attacks effects on the network depends on how difficult or critical is to possess an identity. For example, in a voting system if there is no mechanism to prevent fabricating fake IDs Sybil attacks can change the outcome of the system.

  - *Denial of service attacks:* Denial of service (DoS) attacks aim to prevent the network hosts from accessing the network services. It can happen against services in different layers. For example [Cayirci 2009]:
    - *DOS in Physical layer:* Jamming the carrier to reduce signal to noise below the threshold.
    - *DOS in MAC layer:* Colliding with the CTS signal
    - *DOS in Routing layer:* Wormhole attack, Attacks to deplete nodes' resources, Hello flood attack in which the attacker broadcasts enough hello messages to conceive every node in the network that it is their neighbor.
    - *DOS in Transport layer:* Session hijacking, Jamming acknowledgement, Modifying the sequence number, Replaying acknowledgements in transport layer protocols in which multiple acknowledgements means not successfully message delivery.

- *DOS in Application layer:* Giving false location or power information in localization protocols.

– *Man-in-the-middle attacks:* In Man-in-the-middle attack (MITM) the attacker sits between the sender and the receiver and sniffs the packets sent between them. He may impersonate the sender to send packets to the receiver, or impersonate the receiver to reply to the sender.

Security attacks in multicast routing

Secure multicast MANET routing is of a high importance in hostile environments such as military applications as well as other group based applications such as commercial or voting ad hoc systems. Although the general goals of multicast security is similar to unicast security including to protect secrecy and authentication for legitimate senders and receivers, data integrity, non-repudiation, service availability, privacy and trust management, providing security in multicast routing is more complicated than in unicast.

New issues such as group key management, group membership management, group size, group connection type (tree or mesh based), multiple group networking, group maintenance, one-to-many or many-to-many efficient routing algorithm arise in multicast scenarios. The mentioned goals are even harder to achieve in infrastructure-less environment such as ad hoc and sensor networks compared to infrastructure-based wireless networks such as cellular networks.

The security attacks mentioned above can occur in group-based scenarios as well as unicast networks and affect the normal network operation more or less in the same way. In [Nguyen 2008] the authors studied the impact of several types of security attacks on multicast in MANETs including Rushing attack, Blackhole attack, Neighbor attack and Jellyfish attack. Also in research works such as [Amuthan 2011] and [N.Shanthi 5 09] the effect of attacks such as black hole attack, gray hole attack and worm hole attack are studied on multicast routing performance in MANETs.

2.1.3 Anonymity and privacy threats to ad hoc networks

Identification as an important security requirement of network members prevents Sybil identifiers. Trusted identification is required for guaranteeing the network against Sybil attacks. On the other hand, anonymising the object of interest among the anonymity set opposes the idea of identification. This conflict is discussed in [Martucci 2009] where it is called as Identity-Anonymity Paradox. Moreover, hiding personal information to have data privacy is in conflict to the public need to access information [Cayirci 2009]. However, keeping personal information of network members private in the environment is considered as a critical issue in both wired and wireless networks.

The private information of the internet users either in their profiles or email accounts needs to be protected from disclosure. Wireless networks in particular are more vulnerable to privacy attacks. They are in fact sharing the open wireless links with everybody in the area. The eavesdroppers can use directional antennas to detect transmitted signals and even the path that they traverse. Moreover, mobile ad hoc networks suffer from their limited resources, lack of central supports and poor physical safety regarding privacy protection.

Also, in mobile networks location disclosure is a privacy issue when the network application enters hostile situations. In some applications the privacy of the network members' venue is as important as or even more important than identity anonymity where the adversary is searching to capture the users.

In this section we give an overview of privacy threats to mobile ad hoc networks as summarized in Table 2.2.

Table 2.2: Privacy threats against mobile ad hoc networks

Privacy attacks to ad hoc networks	Location disclosure attacks
	Identification attacks
	Traffic analysis attacks
	Eavesdropping
	Leak of private information in application layer

- *Location disclosure attacks:* This attack is against location privacy of the network members. The attacker is interested in the nodes' venue or the network structure. He tries to find out where each user or his favourite users are located in the network field. After breacking location privacy of the nodes he may compromise them or lunch denial of service attacks or other kinds of attacks against them.

- *Identification attacks:* This privacy threats are against network layer functionalities by identifying the devices or the linkability between two communicating devices, i.e. threatening node anonymity and/or relationship anonymity. The attacker may overhear the routing packets as an external adversary or receive them as an internal one and try to read the source and destination address fields in the packet. For example, the routing protocols AODV [Perkins 1999] and DSR [Johnson 2001] leak the sender and the destination's addresses during route discovery phase.

- *Traffic analysis attacks:* Traffic analysis attacks can happen even when the messages are strongly encrypted. The attacker examines the network traffic pattern in order to infer information about the on-the-fly communication or relationships in the wireless network. He may study the traffic pattern

by detecting the size, the initiation venue, the traversed path and the timing of the message flows and their changes in order to discover the venue of events, functions or owners of the nodes [Cayirci 2009]. One famous countermeasure to this attack is to generate dummy packets such that the traffic has a uniform-like and unchanged pattern all over the network.

An example is that the cluster heads might be detected as the traffic analyser knows that they are busier than the other nodes.

Traffic analysis may occur against different layers:

- Traffic analysis at physical layer

  The attacker senses the carrier and then the traffic rates are studied by him [Cayirci 2009].

- Traffic analysis in MAC and higher layers

  The attacker analyses the MAC frames and data packets and their headers to detect information about routing and relationships [Cayirci 2009].

- *Eavesdropping:* Eavesdropping is actually a physical layer attack and could be against both ID anonymity and location privacy. An strong eavesdropper may establish an eavesdropping network using many colluding unauthorised hostile nodes to intercept and record on-the-fly messages. Wireless networks in general are vulnerable to eavesdropping due to the broadcast nature of their links which can be overheard by any antenna tuned to the corresponding frequency channel. A probable scenario is that the attacker may use the eavesdropped packets' content to determine the identity of the current communication's end-points. In such a case if the packet content is encrypted it will be much more difficult for the adversary to read the packet and he may attempt to decrypt that.

  Ad hoc networks compared to longer range wireless networks are a little more secure regarding eavesdropping since packets are transmitted hop by hop. Therefore, the attacker needs to get close enough to the attacked node to become able to tap the communication [Cayirci 2009].

- *Leaks of private information in the application layer:* The attacker may infer private information about the sender, destination or both by accessing the information encapsulated in the application layer or the data contained in the message payload to identify the communication end nodes or the relationship between the two end nodes. The attacker does not need to be in the radio range of the attacked node and it would be enough to be close enough to a node en route [Martucci 2009]. The authors of

[Tuomas Aura 2008] have analysed the privacy data leakage in the application layer for protocols such as the Domain Name System (DNS) and the Dynamic Host Configuration Protocol (DHCP).

2.2 Privacy and security enhancing technologies

2.2.1 Security enhancing solutions in MANETs

Security in MANETs faces special difficulties as they suffer from lack of a central management to monitor the network members' behaviour. Therefore, the security management in MANETs should be distributed among the mobile wireless nodes. The countermeasures against security threats in ad hoc networks can be of two categories: Intrusion Prevention (defending pro-actively before the attack occurs) or Intrusion Detection (defending reactively after the attack occurs).

2.2.1.1 Proactive defences

- *Physical layer defence:*

  Frequency Hopping Spread Spectrum (FHSS): Switching the carrier among many frequency channels following a pseudorandom sequence agreed in advance between the transmitter and the receiver. FHSS signals are resistant to narrowband interference of Jammers. FHSS transmissions are also difficult to intercept by eavesdroppers as it is supposed that the adversary is not aware of the frequency sequence.

- *Data link layer defence:*

  - Using traffic cover mode: A sudden change in the traffic pattern means that a special event is taking place. By hiding the changes of the traffic pattern the potential passive or active attacks can be prevented.

  - Dynamic mix methods: This technique hides the source and/or destination of transmissions or the source-destination relationships by routing the messages through a chain of numerous mix nodes in the network.

- *Network layer defence:*

  Trust management, source authentication and message authentication: to prevent routing packet injection or modification by hostile nodes. We shortly describe main ideas of few secure routing protocols for MANETs as follows.

  SEAD (Secure Efficient Ad hoc Distance vector routing protocol)

  SEAD [Hu 2003] as a secure ad hoc network routing protocol based on the design of the DSDV (Destination-Sequenced Distance-Vector) routing pro-

tocol deals with attacks that modify routing information broadcast during the routing information updates phase.

SEAD employs the use of a one-way hash chain to authenticate hop counts and sequence numbers to prevent malicious nodes from modifying them. When a node joins the network it generates a one-way hash chain in groups of m (m-1 is the upper bound of the network diameter), h_0, h_1, ..., h_n such that n is devisable by m and $h_i = H(h_{i-1})$. The node will use a mechanism to distribute its authentic hash value h_n among the network members to be used for metric authentications in routing table updates later. For sequence number i and metric j the $j+1$th element from the hash chain h_{km}, h_{km+1}, ..., h_{km+m-1} , i.e. h_{km+j}, is used for metric authentication, where $k = \frac{n}{m} - i$.

When a node sends an entry in its routing update for itself, it will set the sequence number to its own next sequence number and the hash value to the first element of its own hash chain corresponding to that sequence number. When the entry is about another destination, the node will use the sequence number and metric corresponding to that node in its routing table and he will hash the hash value of the received routing update and will include it as the hash value in that entry. This hash value can be authenticated by the nodes that receive this routing update since they have an already authenticated element of the same hash chain [ARGYROUDIS 2005].

Such a mechanism prevents malicious nodes from decreasing metric values in the routing update entries because of the one-way property of the used hash functions.

ARAN (Authenticated Routing for Ad hoc Networks)

ARAN [Sanzgiri 2005] achieves authentication, integrity and non-repudiation of signalling packets. This secure ad hoc routing protocol protects the network against impersonation and repudiation attacks by using predetermined cryptographic certificates issued by a trusted certificate server for end-to-end authentication.

Every route discovery message is signed by its source nod's private key and includes the destination address, the certificate, a nonce and a timestamp. The node who receives such a packet validates the previous node's signature, replaces the certificate and signature with its own and forwards the message.

SAR (Security-aware Ad hoc Routing protocol)

SAR [Yi 2002] provides defence against blackhole attacks. A trust level field is embedded into the route request packets by the source node. When an intermediate node receives the route request packet, it will proceed with the packet only if it meets the required trust level and the packet will be dropped otherwise. If a path from the source to the destination cannot be find with the current required trust level the source nod may decrease the trust metric in its next try.

Therefore, using protocols like SAR the malicious nodes have little chances to become a part of the route and blackhole attacks are very less likely to happen.

SAODV (Secure Ad hoc On-demand Distance Vector Routing)

SAODV [Zapata 2002] adds security extensions to AOVD. This approach secures AODV packets using digital signatures and hash chains. A one way hash chain is used to secure the hop-count information and digital signatures are used to authenticate other fields in the RREQ and RREP messages. SAODV assumes the existence of a asymmetric key management system in the network.

The source of a RREQ or a RREP message generates a random number and sets the max-hop to the ttl field of the IP header. The node sets the hash field with the random number as well as the identifier field of the hash function. Finally, it calculates the top hash by hashing the random number ttl times. This algorithm allows the receiving nodes to verify the hop count of each message by applying the hash function ttl-i times to the value in the hash field. Since every intermediate node applies the hash function once to the hash value before relying it, if the result and the top hash field are the same the hop count is verified. After verification the node applies the hash function on the hash field and forwards it [Fonseca 2006].

Every fields but the hop count and the hash field are signed by the sender and the signatures are modified by every intermediate nodes [Cayirci 2009].

- *Transport layer defence:*

  - Authenticating and securing end-to-end communications through data encryption

- *Application layer defence:*

  - Application layer firewalls can be used as the countermeasure to application layer attacks. They can prevent unwanted traffic from reaching the protected nodes by filtering the packets.

- Cooperation enforcement mechanisms such as credit based mechanisms

2.2.1.2 Reactive defense: Intrusion detection systems (IDS)

Although the secure communication rules, as briefly described above, are exploited in ad hoc networks, due to the lack of a central support as well as the open-air transmission media and the hop-by-hop communication manner, controlling the behaviour of the nodes is necessary. For example, a node may participate in routing process and then in data forwarding phase it can drop the packets quietly.

Intrusion detection deals with monitoring the network to detect any abnormal behaviour in the system to identify and possibly response to the ongoing attacks in the network. Intrusion detection is considered as the second line of defence in securing ad hoc communication. The IDS systems can be designed to detect abnormality, misuse of the network services and operation-specification deviations.

In abnormality detection IDS systems the idea is to save the normal behaviour of the network members to detect any abnormality in the network by comparing the nodes' behaviours to them. In misuse detecting systems, the attacks are identified by comparing the intrusions to the known attack patterns. In specification detection IDS systems a set of expected operations of the used protocol are kept and any misbehaviour in the network compared to them is detected [Cayirci 2009]. Abnormality detection can automatically discover unknown potential attacks although it is subject to a high volume of false positives.

The intrusion detection systems in ad hoc networks should not decrease the network performance and should consume minimum resources to function.

Watchdog and Pathrater

Watchdog and Pathrater is a security approach based on detection and recovery. Watchdog buffers the transmitted packets and monitors if the next node in the path forwards the data packets by listening to it. If so the packet will be removed from the buffer. If the packets forwarded to a neighbor stay in the buffer for more than a threshold number that neighbor will be reported as a misbehaving node.

The pathrater uses the results of watchdog and chooses the most reliable path for packet delivery. The misbehaving nodes are rated negatively. When a route trasfers data packets successfully the nodes belonging to it are rated positively and vice versa. The rate of a path is calculated as the average of its nodes. The most reliable path would be the path with the highest rate among the available routes from the source node toward the destination [Cayirci 2009].

2.2.2 Anonymity on the internet

Privacy have been a challenging topic for many years even in wired networks. Privacy in Internet concerns user anonymity and communication privacy, i.e. not to reveal the identity of the users to their communication peers and to conceal the users information as well as the communication contents from potential attackers.

Mix-net

Privacy solutions initiated from David Chaum's Mix-net idea [Chaum 1981a]. Chaum introduced mix-nets to be used for anonymous email transmission. Each user has a global view of the network topology and transmits its data packets to other hosts making anonymous connections through a sequence of MIXes instead of making direct socket connections to the destination [Boukerche 2006].

Pieces of encrypted e-mails padded to the same length are aggregated at mixer network entities called Mixes. A Mix is a node who stores a number of fixed-length messages from different sources, performs cryptographic operations on them and then sends them out in a random order such that for outsiders it is not possible to distinguish which input message belongs to which output message. A Mix only knows sets of sources and destinations but no special correlation between them. In fact, each Mix waits to receive a number of messages from different sources to mix them up, so this protocol can introduce a high delay specially when the network is under-loaded. To solve this problem dummy packets are added to Mix-nets to speed the protocol up.

A set of Mix devices in a network (Mix-net) is able to provide anonymity even when some of the mixers are compromised, because a single mix entity is sufficient to guarantee anonymity. Mix-nets can be used not only for mixing email messages but for other types of data according to the application.

DC-net

Dining cryptographers networks or DC-nets [Chaum 1988] are privacy preserving primitive proposed again by David Chaum based on binary superposed sending. In DC-net the anonymity set is composed of all potential senders. Each sender shares a random secret key at least with one other user. If sender A is wishing to send a message, it should superpose the message with its exchanged secret keys. Other users superpose in the same manner (if they have no message to send they superpose zero with their shared keys). All messages are transmitted to the receiver. The sum of these messages is just the message of A, because every secret key is added twice and cancelled. Therefore, a message can be delivered without revealing the originator.

However, obviously DC-nets scale very poorly when the number of participants increases. No implementation of DC-net privacy systems is reported in the literature so far [Sirer].

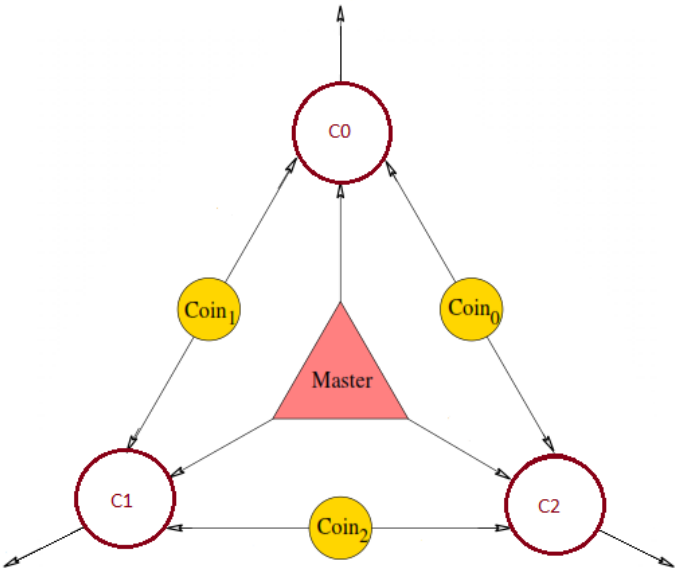

Figure 2.2: The Dining Cryptographers protocol

The story behind Dining cryptographers is interesting. Suppose that three cryptographers are dining together with their master. At the end of the dining the Master will decide who among himself and the three cryptographers should pay. Now suppose that the cryptographers would like if one of them or the Master himself has paid for the dinner and in the case that one of them has paid they wish to keep his ID anonymous.

The Dining Cryptographers protocol solves this problem providing a distributed anonymous system. Each cryptographer tosses a coin and can see also the result of his left hand side coin. Then any non-payer cryptographer would say *Yes* if the result of the two coins that he are aware of are the same and would say *No* otherwise. The paying one, if there exist any, would announce the opposite. It can be shown that if the Master is paying the number of *No* answers is even and otherwise one of the cryptographers has paid for the dinner. Therefore, from the answers it will be clear if the payer has been the Master or not and the ID of the payer if he is one of the cryptographers is not revealed [Chatzikokolakis 2007].

Crowds

Another privacy solution for wired network server requests is Crowds proposed by Reiter and Rubin in [Reiter 1998]. Crowds is a randomized packet forwarding network. Each user joins a group of users by registering itself at the Blender

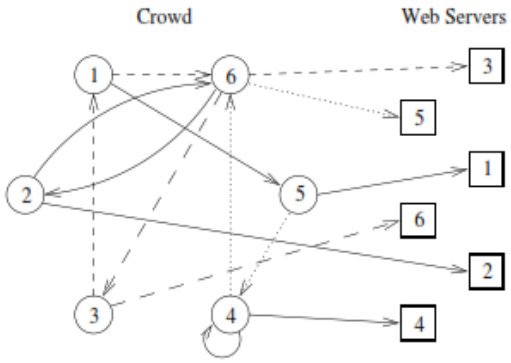

Figure 2.3: Examples of paths in Crowds (The source and the web server of each path are given the same number)

which is a special server in the network used for group membership and key distribution. A server request is not sent directly to the server but chained through a random number of other group members. One of the nodes will decide by a given probability to forward the request to the server. The server will not be able to determine the original sender of the request but can send answers back using the tunnel to the source. In this manner, no single point of failure compromises the sender's anonymity.

Figure 2.3 ([Reiter 1998]) shows examples of connections made to the web servers using Crowds.

Since the appearance of the message is not changed, a raffic analysiser still would be able to trace the traffic flow of each connection by eavesdropping on the path. Crowds provides sender privacy but not receiver privacy since the message is sent to the end server directly from the last user in the forwarding chain.

Onion routing (OR)

Onion Routing is one of the most famous anonymous communication technologies. In Onion Routing the fact that who is communicating to whom remains hidden from both outsider observers and insider compromised node. An onion is a layered data structure transmitted along the route including the data as a core and the properties of the connection at each point [Goldschlag 1999]. A set of onion routers are selected by the initiator connection in set-up phase. The initiator encrypts the data in an onion structure using the public keys of the

decided sequence of onion routers and sends it to the first one.

The onion will be forwarded along the anonymous route from one onion router to the next one. Each onion router peels away an encryption layer using its private key. Also every router pads the onion to keep its size the same along the route. Therefore, the onion will look different at each hop and cannot be tracked en route. Also compromised onion routers cannot cooperate by correlating the data stream that each one can see. Each onion router can only identify the previous and next router in the route. Finally, the recipient will receive the data as plaintext since all the layers are removed by the onion routers en route. As a result, the intermediary nodes are prevented from identifying the sender, destination and the content of the message. Such an anonymous route can be used in the reverse direction as well.

Onion routing as described above is vulnerable to timing analysis because an attacker may be able to link the corresponding incoming and outgoing onions by observing the time of receipt and send of packets at routers.

Tor [Dingledine 2004] is the next generation onion router which many improvements over the old onion routing design.

2.2.3 Privacy techniques in MANETs

The privacy enhancing protocols proposed for wired or infrastructure-based networks are not suitable for ad hoc or sensor networks. Such protocols are designed based on the constant presence of a centralized support and/or constant network traffic flow which conflict with the requirements for mobile ad hoc networks. To protect the information about the sender or receiver identities, locations or relationships some solution has been proposed for infrastructure-less networks.

Packet flooding can be used to cover the current radio activities with too many transmissions to hide the object of interest. Flooding has been basically proposed for sensor networks and can be performed in different ways:

- Baseline flooding: Every node broadcasts every received packet.
- Probabilistic flooding: Every node broadcasts every received packet with a specific probability.
- Fake packet flooding: Random nodes broadcast fake packets in the same way as real ones.

As described before one important privacy attack in MANETs is traffic analysis. To cope with traffic analysis, the main solution is de-patterning the traffic flow. Fake data messages can be used to simulate the real network activities to mislead the attacker. De-routing data packets or changing the timing of messages, e.g. to buffer a number of packets and send them out together with random delays, are other de-patterning privacy solutions.

In this thesis we will propose a solution to location privacy issue by hiding the packets' type misleading the adversary about the origin of that packet. Packet de-routing techniques will also be used in our work. Moreover, we will propose a privacy technique that changes the timing of kind of messages to prevent timing analysis in multicast routing.

Most of the current anonymity and privacy enhancements are designed as anonymous routing protocols although some of them could be designed as overlay anonymity mechanisms between the application and the transport layer [Martucci 2009]. Different privacy levels can be achieved by different privacy enhancing protocols. Moreover, some anonymous routing protocols consider a global passive attacker while the others consider a limited eavesdropping ability for the attacker. Measuring the privacy level when the privacy enhancing solution and the adversary's model are given is another challenge to evaluate the contribution of privacy solutions as studied in section 2.2.3.2.

In the following section we describe few MANET routing protocols and then we will review privacy measurement tools.

2.2.3.1 Anonymous ad hoc routing protocols

Anonymity techniques used in wired networks are not directly suitable for wireless networks. Eavesdropping is much easier to be performed against wireless networks compared to wired networks, because no physical access to the network is required. Furthermore, as described before privacy is much more challenging in MANETs due to their special vulnerabilities and constraints.

During last decade there have been several anonymity and privacy protocols proposed for wireless ad hoc networks mostly designed in routing layer and trying to address the identity anonymity issue.

In this section, we describe the main points of several ad hoc routing protocols proposing anonymity and privacy protection solutions.

ANODR (ANonymous On Demand Routing with Untraceable Routes for Mobile Ad Hoc Networks)

ANODR is the first on demand anonymous routing protocol proposed for ad hoc networks. ANODR is identity free [Kong 2007], and uses a route pseudonymity approach in which each node on the route is associated with a pseudonym. Since the nodes identities including the source and the destination's IDs are not used in the routing process therefore the nodes' IDs and their locations would be unlinkable. ANODR provides the property of route anonymity, which means the adversaries cannot trace the nodes on a route in terms of their identities although the eavesdropper attackers may be able to trace the route venue.

Also a single point of compromise does not compromise location privacy of other legitimate nodes on the route, so an on demand ANODR route is traceable

only if all forwarding nodes along the route are intruded.

The main cryptographic tools used in ANODR include the idea of layered encryption of MIX-Nets [Chaum 1981b] for message flows and trapdoor opening. In MIX-Net any message sent from the sender to the receiver can be decrypted only by the receiver, for example using the receiver's public key. If a message should be sent from S to D via Mixes A and B, the input of the Mix-Net would be as $\{B, N_S^2, \{D, N_S^1, \{m, N_S^0\}_{PK_D}\}_{PK_B}\}_{PK_A}$, where N_S^is are node nonces. This is the same idea used in *Onion Routing*. Each MIX node on the route will peel away one layer of the constructed onion and forwards the remaining onion to the next node.

Broadcast with trapdoor information is another cryptographic idea used in ANODR to determine who is the intended destination without revealing its ID.

A one way function: A function $f : X \longrightarrow Y$ is a *one way function* if it is easy to obtain the image for every element $x \in X$, but computationally infeasible to find a pre-image for any given $y \in Y$.

A trapdoor one way function: f is a *trapdoor one way function* if it is a one way function, and it becomes feasible to find a pre-image for any given $y \in Y$ given some *trapdoor information*.

Broadcasting with trapdoor assignment prevents specifically identifying the receiver. The trapdoor information embedded in the message, which is a well known tag like predetermined bit-string "You are the destination" [Makki 2007], can be decrypted only by the receiver. The message can be delivered just to the receiver.

ANODR protocol exploits both symmetric and asymmetric cryptography operations as will follow.

- ANODR Protocol Design:

- *Anonymous Route Discovery*

  The sender initiates a one time RREQ packet containing an anonymous global trapdoor, an onion and a sequence number. Since RREQ flood and public key cryptography are both expensive procedures, ANODR uses symmetric key cryptosystem to construct the onion. Each intermediate node, x, receiving the RREQ packet first checks if a RREQ packet with the same sequence number has not been seen before. If so it would add a self-aware layer to the onion encrypting the incoming onion with its own symmetric key, K_x [Kong 2007]. Each intermediate node after forwarding the RREQ, tries to open the global trapdoor to check whether it is the destination. When any intermediate node decrypts the trapdoor it would see just some random bits, but only the attended destination will see the well known tag after decryption. The sender uses the public key of the destination as the global trapdoor key in the first RREQ packet from the source to the

destination. The public keys of the nodes are known from some credentials published by a well known offline authority, ψ. The credentials are in this format: $[id, PK_{id}, validtime]_{SK_\psi}$. To improve the performance, for the next RREQ messages from the same sender to the same destination a symmetric key would be used, which is piggybacked in the first global trapdoor from the destination to the sender [Makki 2007].

Eventually the destination will receive the onion and will use that to deliver a RREP packet back to the source. The RREP packet includes the onion, the proof of global trapdoor opening ($Proof_{des}$) to show the source that it has successfully opened the trapdoor and a random route pseudonym. When the source node receives the RREP packet and reveals the onion core it sent out a while ago, the signalling procedure ends and the anonymous virtual circuit establishing (storing the route pseudonyms) is done during the RREP packet flood.

This is very hard for cryptanalysts to find out the relation between incoming and outgoing onions in order to correlate the route pseudonyms established on top of the cryptographic onions. But an unbounded eavesdropper can infer some other information from the RREQ and RREP packets. RREP packets with the same $Proof_{des}$ might belong to the same route. Also, RREQ packets with the same sequence number and global trapdoor are on the same route. Such eavesdropper can record all onions during RREQ phase, then the RREP packets using the onions from previously matched RREQ packets belong to the same route. To address this problem each RREQ forwarder node (upstream node in RREQ phase) should embed its one time public key from a one time public/private key pair into the RREQ packet, and then in the RREP phase the upstream node in RREP phase (who was the downstream node in RREQ phase) will use that to encrypt the onion and the $Proof_{des}$. The one time public keys are generated in the idle time of the nodes. The length of these keys should be minimized to reduce transmission overhead, but should also be long enough to prevent crypanalysists from their threats. So the RREQ and RREP packet are in these formats:

$$< RREQ, seq\#, global - trap, onion, PK - 1time >$$

$$< RREP, \{K_{seed}\}_{PK-1time}, f_{K_{seed}}(Proof_{des}, onion) >$$

In which K_{seed} is the route pseudonym, which acts also as a secret key shared between two consecutive RREP forwarders and is described as follows [Kong 2003], [Kong 2007]. As the RREP packet is flooding, the onion is decrypted by the intermediate nodes that added an encryption layer to that in the RREQ phase. In RREP phase each upstream node chooses

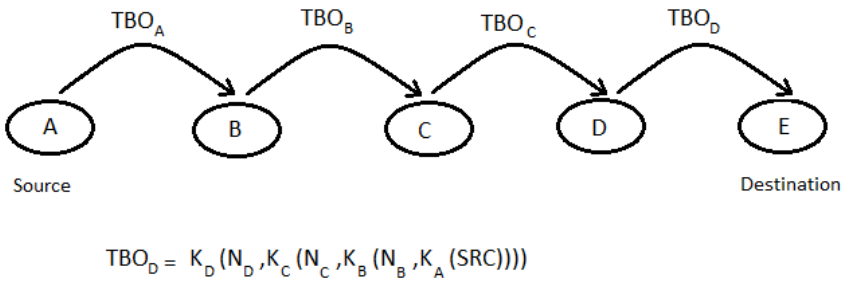

Figure 2.4: Anonymous route discovery using Trapdoor Boomerang Onion (TBO) (in RREQ phase)

a random number as a *virtual circuit identifier (VCI)* that is the route pseudonym. Each downstream node can extract that using its one time public key, $PK - 1time$, and then use that to find out the onion. Then the downstream node will do the same function as the previous one, and overrides the same field in the packet with its own randomly chosen route pseudonym. Each intermediate node receiving the RREP packet processes its route table recording the correspondence between its own VCI (VCI_D) and its upstream node VCI (VCI_U), in this format [Makki 2007]:

$$VCI_U \rightleftharpoons VCI_D$$

The anonymous route discovery phase using Onion is shown in Figure 2.4.

- *Anonymous Data Forwarding*

  Now that an anonymous virtual circuit is established between the source and the destination, the data would be delivered to the recipient using the route pseudonyms. Data packets include the data and the next link pseudonyms. Every node receiving the data packet should look up the data packet route pseudonym in its routing table entries, *Incoming VCI* $\rightleftharpoons$ *Outgoing VCI*. If the route pseudonym of the received data packet does not match any incoming VCI in its table, the packet is not intended to this node and will be discarded. Otherwise, it will change the route pseudonym field to the matched outgoing VCI, and then broadcasts the modified data packet. The packet will finally arrive at the destination [Kong 2007]. The data packet format is as follows.

$$< DATA, Route\ pseudonym, F_{K_{seed}}(index, payload) >$$

Where index is the sequence number of the data packet. The route pseudonym is used as a secret key to encrypt the payload and the data index.

One important property of ANODR is *self-synchronized route pseudonym update*. As long as the route pseudonym shared between two intermediate nodes changes to the same value, the two nodes can use that constantly. It is useful to address a traffic analysis and replay attacks. If the data packets appear computationally one-time over the links the observers will not be able to trace them along the whole route. ANODR uses the shared VCI K_{seed} as the secret seed to generate cryptographically strong pseudorandom sequences of route pseudonyms. The sequence will be used as the route pseudonyms to transmit consecutive data packets [Kong 2007]. The i-th route pseudonym is generated by applying a fast one-way function on K_{seed} and then feeding the output back to the input repetitively for i times [Kong 2003].

$$N_i^k = \underbrace{f(f(...f(N_i)...))}_{\text{k}} = f^k(N_i)$$

N_i^k is the kth route pseudonym of the link.

One disadvantage of using ANODR is the public key crypto-systems used in RREQ flood and in RREP unicasts between each pair of intermediate nodes which causes an extra overhead.

SDAR (Secure Distributed Anonymous Routing)

SDAR [Boukerche 2004] is an on demand anonymous routing protocol for MANETs that uses a proactive neighborhood scan at each network member. Every node gathers its neighborhood information periodically. SDAR is designed based on a distributed malicious behaviour detection to improve the reliability and security of the routes [Boukerche 2004].

- Trust Management System

  The trust management system used in SDAR functions based on every nodes' community consisting of the node itself as the central node and its one hop neighboring nodes. Each node frequently broadcasts a *HELLO message* including its public key to their neighbors. In every community the central node listens to its neighbors' hello messages and records the public key of each neighbor nodes. Such records are regularly updated. The members of a community are classified into three trust levels obtained

and updated according to their behaviour. The central node generates two group keys referred to as *High Trust Level Community Key (HTLCK)* and *Medium Trust Level Community Key (MTLCK)* and the routing messages are encrypted with a key according to the required trust level of the source. The MTLCK is shared with the medium trust level nodes and both keys are shared with the highest trust level nodes. A received routing message is propagated by the central node only to its community members that meet the source node requested trust level. The trust level of a node will decrease if it do a malicious behaviour [Boukerche 2004].

- Malicious behaviour detection

  If a node lunches *Malicious Dropping* or *Malicious Modification* attacks its neighbors would identify its malicious behaviour by overhearing it [Boukerche 2004]. The trust level of such a node will degrade at its neighboring nodes and it will affect its participation in the routing process.

- Protocol Design

  The routing is performed in three phases: *Path Discovery Phase, Path Reverse Phase and Data Transfer Phase.* First the source node broadcasts a route request message including a temporary one time public key (TPK) and a secret key K_s. The TPK acting as a unique identifier of the route request message will be used by the intermediate nodes to encrypt routing information and K_s can only be seen by the receiver to encrypt the data as well as to protect against replay attacks. The private key corresponding to TPK, TSK, is embedded in the path discovery message encrypted with K_s. Every intermediate node records TPK, generates a random symmetric key, K_i, encrypts K_i with TPK and appends that to the message. Each forwarding node adds $< SN_{session-ID_i}$, ID of the ancestor node, $K_i >$ to the internal mapping table. The path discovery message just sent by the sender, and The path discovery message just processed by $node_i$ is shown in Tables 2.3 and 2.4 [Boukerche 2004], [Makki 2007].

Table 2.3: Path discovery message sent by the sender

$TYPE$, $TRUST\text{-}REQ$, TPK
$E_{PK_R}\{ID_R, K_S, PL_S\}$
P_S
$E_{K_s}(ID_S, PK_S, TPK, TSK, SN_{Session-ID_S}, Sign_S(M_S))$

Table 2.4: Path discovery message processed by $node_i$

$TYPE$, $TRUST\text{-}REQ$, TPK
$E_{PK_R}\{ID_R, K_S, PL_S\}$
P_S
$E_{K_s}(ID_S, PK_S, TPK, TSK, SN_{Session-ID_S}, Sign_S(M_S))$
$E_{TPK}(ID_i, K_i, SN_{Session-ID_i}, Sign_{ID_i}(M_{ID_i}))$
..
$E_{TPK}(ID_i, K_i, SN_{Session-ID_i}, Sign_{ID_i}(M_{ID_i}))$

The source node determines its required trust level, $TRUST\text{-}REQ$, in the route request message. PK_R and ID_R are the destination's public key and identification respectively. P_S is generated by the source node and used to hide real routing information and to protect against *message size attack* and PL_S is its length.

$$M_S = H(TYPE, TRUST - REQ, TPK, TSK, ID_R, K_S, ID_S, PK_S, SN_{Session-ID_S}, PL_S, P_S)$$

and $SN_{Session-ID_S}$ is a random number generated by the source node and is mapped to the encryption key K_S to use with the message. $Sign_S$ protects the integrity of the message. The destination finds K_S using its private key and then finds TSK using K_S [Boukerche 2004].

The receiver puts all ID's and session keys and the generated random numbers of the intermediate nodes in a new message and encrypts that several times using their session keys and sends that to the first node in the reverse path (in the form of MIX-Net onion). With each encryption, the receiver R adds a layer that contains the random number generated by the node and the random number generated by the node's next-next-hop node along the reverse path to the sender. Each intermediate node that receives the path reverse message uses the $SN_{Session-ID_i}$ to retrieve the key for the session, removes one encryption layer and forwards the message to the next node on the reverse path to the source node. The ID of the node from which the message was received is added to the successor node entry corresponding to the random number into the mapping table. When the source node receives the message, it decrypts the message and passes the information about all the intermediate nodes (i.e. the route) to the higher application.

Now the sender and the receiver both have a symmetric key shared with all intermediate nodes on the discovered path [Boukerche 2004], [Makki 2007].

Data Delivery: the sender and the receiver use layered encryption approach like MIX-Net onion to deliver data payload to each other, and only the sender and the destination are able to decrypt entirely the core of this messages using K_S that is shared between them [Makki 2007].

Each neighbor detection message and neighbor community update is significant regarding overhead. Also RREQ flooding, due to the public key encryptions introduces a noticeable overhead to the protocol. Each RREP and DATA packet includes l onion layers, so these messages are l times a typical symmetric key length which is at least 128-bit long [Makki 2007].

A drawback of SDAR is that the destination learns every intermediate node's identifier. Also every forwarding node should perform a decryption, encryption and signature generation.

MASK

MASK performs routing by establishing virtual circuits between the sender and the receiver . In MASK an off-line Trusted Authority (TA) assigns a large set of collision-resistant pseudonyms to each node and a corresponding secret point set is generated from the pseudonyms set. Nodes use different pseudonyms when moving to a new location. A node does a three stage handshake for exchanging a chain of secret keys and locally unique LinkID pairs corresponding to the pseudonyms used during handshake with each of their neighbors (using a pairwise key agreement proposed in [Dirk Balfanz 2003]).

The route request message contains the current pseudonym of the node forwarding it. Every forwarding node records the pseudonym contained in the route request message it received from the previous hop in its reverse route table.

The destination will initiate a route reply packet when it receives the request. The reply message contains the identity of the destination encrypted with the current session key shared between the recipient and the next node en route which will be replaced at every hop. The link identifier in the reply packet can only be interpreted by the previous node (who the message is intended to) by looking it up in its neighbor table. Every intermediate node uses this link identifier in the reply packet to find out that the packet is intended to it and uses the destination ID in it to find out to whom it should forward the packet. The intermediate nodes store the updated upstream and downstream corresponding link identifiers (based on the received ones) in their forwarding route tables to use for text packet forwarding on the same route.

Data packets are forwarded on the route using the link identifiers and corresponding keys in the forwarding route tables. Re-encryption at every hop is required.

Every two one-hop neighboring node en route update both the shared key and the link identifier whenever a new message is sent over the route [Martucci 2009].

The route request message carrying the real identity of the destination is broadcast all over the network to prevent the observers from linking the ID and the location of the recipient together. Therefore receiver ID anonymity is not protected in MASK but its location privacy is supported partially.

Other security tools used in MASK include random padding, dummy traffic, intentional delaying of communication data and data forwarding through multiple paths [Martucci 2009].

One drawback of MASK is that it requires a tight synchronization for keys or pseudonyms between neighbors.

ODAR

ODAR presented in [Sy 2006] is an on demand anonymous routing protocol using Bloom filters. A Bloom filter is a probabilistic data structure storing the elements of a set with the possibility of testing if one given element is a member of it or not. In ODAR the path information as well as the sender and destination identities are stored in the Bloom filter.

First the source node requests the public key of the destination from a server and generates a session key as its pseudonym and a secure hash value as the pseudonym of the destination. Both pseudonyms are put in a route request message sent to the destination to find a route to him and are only recognizable by the communicating pair.

When the route discovery packet arrives at the destination, it will be able to use its private key and the sender's pseudonym to regenerate the destination's pseudonym to check if it is the intended destination. If so the node would generate a route reply packet which will be forwarded toward the source node by the nodes who are on the source route.

The IDs of the intermediate nodes is hashed into a Bloom filter in the data packets. Each node receiving a packet checks if its ID is embedded in the appropriate Bloom filter. If so the node would forward the packet.

In ODAR the false positive rate of the Bloom filters can lead to unsuccessful route discovery or data delivery.

ARM

ARM, [Seys 2006], is another anonymous routing protocol proposed for ad hoc networks. In this protocol it is assumed that every two nodes, x and y, that may want to communicate to each other share a secret key k_{xy} and a secret pseudonym, Nym_{xy}, which is renewed for every new RREQ. Also, every node x shares a broadcast key with its neighbors, k_{x*}.

The source node generates a public key pair, $priv_D$ and pub_D, for the destination and a secret key, k, and then the RREQ message is initiated of this format

$$< RREQ, Nym_{SD}, TTL, pub_D, id_D, pub_D(n_S, k_S) >$$

Where

$$id_D = k_{SD}[D, k, priv_D], k[Nym_{SD}]$$

TTL is the time to live set by the RREQ source node and is decremented at every forwarding node. Every node forwards the RREQ message only if $TTL \geq 1$. TTL is used to prevent packets from long or endlessly circulating in the network regarding the limited power of the ad hoc nodes. Every node maintains a list of valid pseudonyms shared with other nodes which are updated for every connection and are synchronized between the two nodes sharing them.

Receiving the RREQ message, every node looks up for the Nym_{SD} in its current list of valid pseudonyms. If it is found it means that the node migth be the intended destination, so the node would try to open id_D using the correspondent k_{SD} expecting to see its own identifier, D, in the first part of that. If so the node considers itself as the intended destination of this message and it would generate the RREP message. Otherwise, it should record Nym_{SD} (if it is the first time receiving this RREQ message and it is not recorded already). In the second case the node generates a pair of link identifiers, n_i and k_i, and stores $< Nym_{SD}, n_i, k_i, k[Nym_{SD}] >$ in its routing table. Then the node adds n_i and k_i to the received encrypted link identifiers (the onion) and encrypts that using pub_D found simply in the message. Then after decrementing the TTL it broadcasts the result RREQ message to its neighbors.

$$< RREQ, Nym_{SD}, TTL, pub_D, id_D, pub_D(..., pub_D(pub_D(n_S, k_S), n_i, k_i), ...) >$$

The RREQ message is forwarded till it reaches the destination. The destination, beside initiating the RREP message, forwards the RREQ message just like the intermediate nodes pretending not to be different from them. When the TTL reaches zero the nodes stop forwarding the RREQ packet.

When the destination decrypts the id_D successfully, $priv_D$ would be extracted which is used by the destination to open the onion, $pub_D(..., pub_D(pub_D(n_S, k_S), n_i, k_i), ...)$. Then it uses the extracted link identifier pairs to construct the onion for the RREP message. It generates a random TTL and initiates the RREP message of the following format

$$< RREP, k_{D*}[Nym_{SD}, TTL], k_n[n_n, k'_n, k, k_{n-1}[n_{n-1}, k'_{n-1}, k, ..., k_S[n_S, k]]] >$$

Receiving such a message every intermediate node checks if it has forwarded a RREQ message with the same Nym_{SD} a while ago or not. If so and if it is the first time the node sees this RREP message it tries to decrypt the onion using its stored k_i. If the corresponding stored n_i appears at the beginning of the decrypted onion it realizes that it is on the discovered anonymous route. Then

it uses the retrieved k to decrypt the stored $k[Nym_{SD}]$ (from RREQ message) and check if the result is equal to Nym_{SD}. If so the node would generate a new random TTL and encrypt the header with its broadcast key, and broadcast the modified RREP message. Also the node stores $(k'_i = h(k_{i-1}), h(k_i))$ in its routing table which will be used as its secret key shared with the previous and next hops on the route.

In the case that the intermediate node has not forwarded the corresponding RREQ message before or it is not the first time receiving this RREP packet it would replace the onion with a random one, decrement the TTL, decrypt the header with its own broadcast key and broadcast the modified message. Therefore the RREP message will be forwarded over some fake routes around the real route and the adversary cannot distinguish the real discovered route among the cloud of possible routes.

2.2.3.2 Privacy measurement

Privacy as a property of anonymous systems needs to be quantified to provide a measure to evaluate or compare the degree of privacy provided by the system. On the other hand it is not a trivial task to find a straight-forward and perfect tool to measure the privacy of an object. There have been several mathematical tools proposed to be used to measure anonymity. The most famous privacy metric is Shannon entropy that can be seen as the uncertainty level of the observer to recognize the hidden element among the anonymity set. In this section we give an overview of the state-of-the-art of anonymity measures. Then we will propose a new metric for this purpose.

Reiter and Rubin qualified privacy levels in [Reiter 1998] by defining different privacy strength degrees as follows:

- Absolute privacy: The message transmission is not observable to the attacker.

- Beyond suspicion: The attacker can observe the sent message but the real sender is the same likely to be the originator of the message as any other member of the anonymity set.

- Probable innocence: From the attacker's point of view, the real sender could be the originator of the message with the probability of fifty percent.

- Exposed: The attacker can identify the source of the message.

- Provable exposed: This is the lowest level of sender anonymity in which the attacker can identify the source node and also is able to prove its identity to others.

Privacy quantification means to define a metric to measure the privacy degree of the anonymous object. A metric is a system or a standard of measurement. A privacy metric for a network is a measurement standard to quantify levels of privacy to numeric values [Ma 2011].

Theoretical approaches

- *Anonymity set size*

  In 1981 Chaum proposed to measure anonymity in a very simple way by considering only the anonymity set's size in [Chaum 1981b]. It means to quantify the privacy based on the total number of network users who are indistinguishable to the adversary with respect to a specific action in the system. For example, the number of users suspected to be the sender or the receiver of a specific message. When more users are considered to belong to the anonymity set, the observer would be more uncertain about the actual object of interest. So the bigger the anonymity set size, the higher the privacy level. If the adversary knows that there are N users in the whole network and he compromises or eliminates C agents among them during the attack the level of privacy of the object of interest will be $N - C$ [Douglas J. Kelly 2008].

  Although the anonymity set size is potentially an important factor in determining the anonymity level but it is not the only factor. The adversary's knowledge might be more than the number of suspected elements as he may infer that some members of the anonymity set are more likely to be the anonymous object, as a traffic analyser.

- *Shannon entropy*

  There are information theoretical approaches to quantify anonymity mostly based on concept of entropy proposed in [Claudia Diaz 2002] and [Serjantov 2002]. The uncertainty of the adversary about the anonymity set regarding how likely each member is to be the wanted object is directly related to the privacy level of such an object. Consider the discrete random variable X with N elements with the probability distribution of $P(x_i)$ assigned to x_i. The entropy H(X) of this probability distribution can be calculated as

$$H(x) = -\sum_{i=1}^{N} P(x_i) log_2(P(x_i)) \tag{2.1}$$

  This value actually models the amount of the observer's uncertainty about the result of the random event. In a communication system it can model the adversary's uncertainty about a private object. For example, suppose

that a traffic analyser trying to discover the originator of a message ends up with a probability distribution on a set of specified users in the network. The entropy of the probability distribution gives the traffic analyser's uncertainty about the source node and can be interpreted as the amount of additional information, in terms of number of bits, that the attacker needs in order to definitely identify the message's originator.

The maximum entropy can happen when all of the anonymity set member's are assigned the same probability, i.e. $\frac{1}{N}$. Obviously the maximum entropy would be calculated as

$$H_{max}(x) = log_2(N) \tag{2.2}$$

Diaz et al [Claudia Diaz 2002] presented the anonymity level normalized to the maximum entropy, i.e. they defined the observer's uncertainty about the anonymous object relative to the situation that all anonymity set member's are equally likely to be the object of anonymity. The degree of anonymity in this approach would be calculated as follows.

$$d = \frac{H(x)}{H_{max}(x)} \tag{2.3}$$

Since this metric compares the entropy to the maximum value of it, the result would range from zero to one.

- *Local anonymity*

  Authors of [Gergely Tóth 2004] suggested a very simple definition for the privacy level based on the highest assigned probability to the users.

  $$\theta = Max_{1 \leq i \leq N} p(i) \tag{2.4}$$

  They believed that θ is more important than d from the perspective of a user. The relationship between θ, H and d is given in [Gergely Tóth 2004].

  It is obvious that in any probability distribution set we always have:

  $$d \geq -log_N(\theta) \tag{2.5}$$

- *Isolation factor: privacy as a three-dimensional value (Shannon entropy, local anonymity and isolation factor)*

  The authors of [Neeraj Jaggi 2011] argued that neither Shannon entropy [Claudia Diaz 2002] nor local anonymity [Gergely Tóth 2004] are perfect metrics to measure privacy. They presented few examples showing the weakness of both mentioned metrics in giving the expected result

in those given cases. [Neeraj Jaggi 2011] proposes a new metric called $Isolation factor$. An element in a probability distribution set is considered *isolated* if its probability is noticeably higher compared to the other elements. To calculate the isolation factor one needs to detect the isolated probabilities in the set. A user with an isolated probability is called an *outlier*, i.e. a user whose probability from the attacker's point of view deviates considerably from the rest of the probabilities. There are multiple approaches to detect *outlier*s in a random set. [Neeraj Jaggi 2011] used the one presented in [Peirce 1852] and [Gould 1855] which is called Peirce's criterion. In the first iteration the algorithm tests to approve or reject whether there is at least one *outlier* in the system. If the result of the test is positive the algorithm is repeated to test if there are at least two *outlier*s in the set. This procedure will continue till the result of one iteration is negative. The last iteration determines the number of *outlier*s.

After detecting the number of *outlier*s one needs to calculate the *isolation factor* using the following formula:

$$\tau = \frac{\sqrt{\sum_{i=1}^{N}(p_i - \bar{p})^2} - \sqrt{\sum_{i=1}^{N}(p_i - \bar{q})^2}}{max(\zeta, 1)} \tag{2.6}$$

where ζ is the number of outliers and τ is the *isolation factor*. $\bar{p} = \frac{1}{N}$ is the mean of probability distribution and $\bar{q}$ is the mean of probability distribution without the outliers in the set (i.e. $\bar{q} = \frac{\sum_{i=\zeta+1}^{N} p_i}{N-\zeta}$ when the *outliers* are the first ζ elements in the set). Therefore, *isolation factor* gives the impact of each outlier in increasing the standard deviation of the distribution.

For any distribution we always have

$$0 \leq \tau \leq 1 \tag{2.7}$$

The authors proposed to use a combination of d, θ and τ to calculate the amount of privacy applying a weight on each one. They consider the three-dimensional metric of (d, θ, τ) as the privacy level of the system.

$$R = \sqrt{W_d \times d^2 + W_\theta \times (1-\theta)^2 + W_\tau \times (1-\tau)^2} \tag{2.8}$$

such that $0 \leq W_d, W_\theta, W_\tau \leq 1$ and $W_d + W_\theta + W_\tau = 1$

The authors argue that these weights should be selected according to the attributes of the system. If an end user is only interested in its local anonymity he would view the system using weights $W_d = W_\tau = 0$ and $W_\theta = 1$. If we consider the attacker's perspective the weights should be

chosen close to $W_d = W_\theta = 0$ and $W_\tau = 1$. From the system designer's perspective the weights would be close to $W_\tau = W_\theta = 0$ and $W_d = 1$.

Obviously R lies in the range of $[0, 1]$. R gets close to its maximum value when the probability distribution is uniform, i.e. $p_{i_{1\leq i\leq N}} = \frac{1}{N}$ where $d = 1$, $\theta = \frac{1}{N}$ and $\tau = 1$. As $N \to \infty$, $\theta \to 0$ and $R \to 1$. We note that for anonymity set size of N the maximum values of τ and R depend on N and are less than 1.

On the other hand, for the probability distribution of $p_1 = 1$ and $p_{i2\leq i\leq N} = 0$ the parameters are changing as $d = 0$, $\theta = 1$ and $\tau = \sqrt{\frac{N-1}{N}}$. In this case again if $N \to \infty$ then $\tau \to 1$ and $R \to 0$.

- *Relative entropy*

  Relative entropy is a tool to compare two probability distributions. It gives a sense of the colloquial distance between the probability distributions. The relative entropy $D(p\|q)$ between the two probability distributions p and q is defined as:

$$D(p\|q) = \sum_{i=1}^{N} p_i log_2(\frac{p_i}{q_i}) \tag{2.9}$$

  Relative entropy is not necessarily symmetric, i.e. $D(p\|q) \neq D(q\|p)$. $D(p\|q)$ is always positive and is zero if and only if the two distributions p and q are the same. It has some applications in coding theory as well as gambling theory to measure the bad effect of designing a code or gambling strategy supposing the distribution to be q while it is actually p. [Yuxin Deng 2007] proposes a privacy metric based on relative entropy. This work measures the anonymity level as the distance between the probability distribution and the one resulted after permutations of anonymous actions. The authors beleive if the maximum value of this distance is zero the prvacy level is very high and if it goes to infinity the privacy level is zero.

- *k-anonymity*

  k-anonymity [Sweeney 2002] is a metric based on anonymity set size. k-anonymity originally was developed for database systems as a metric for data released. The structure of the data release is a table with tuples in the rows and attributes in the columns. Attributes are sensitive or non-sensitive information about the tuple that could be used alone or in combination with other attributes to identify the corresponding tuple, called quasi-identifiers.

  Sensitive attributes could lead to significant information leakage. A set of indistinguishable tuples with respect to specific identifying attributes is

called an equivalence class and corresponds with the anonymity set regarding those attributes. The aim in k-anonymity is to have at least k -1 other indistinguishable tuples with respect to a certain set of quasiidentifiers for each tuple. For example in location privacy context k-anonymity means that the given location information of one user is indistinguishable from the location information of at least $k - 1$ other users [Ma 2011].

Chapter 3

Privacy in unicast ad hoc routing

In this chapter we propose a protocol to provide receiver location privacy in mobile ad hoc networks. In general, anonymity is achieved by hiding the object of interest among a number of entities, the *anonymity set*, so that it is not obvious to any outsiders which anonymity set member is the real entity. What we will achieve in designing this protocol is to provide the destination's location privacy against a global eavesdropper. Location privacy attacks can be performed by traffic analysis to detect the network elements' venue observing the network traffic patterns.

We propose a solution in this chapter in which the routing is performed in a way that the location of the destination node cannot be inferred by the adversary. We will study the privacy properties and the routing performance of the proposed approaches afterwards. Non of the existing anonymous MANET routing protocols investigates location privacy for the destination node against a strong eavesdropping adversary.

3.1 Introduction

Infrastructure-less networks including ad hoc and sensor networks allow the users to spontaneously establish local wireless communication. Although users can save their time and costs by deploying an ad hoc network, the wireless nature of communication and the lack of a central entity to monitor the nodes' behavior causes MANETs some special security and privacy vulnerabilities. In hostile environments the ad hoc communication is susceptible to link attacks including passive eavesdropping and active attacks such as impersonation, packet replay and distortion. Furthermore, the nodes roaming in a hostile environment sometimes might be under the risk of being captured by the adversary. A compromised node should be considered as an internal adversary.

On the other hand, spoofing the nodes' private information such as their identities and secret keys, tracking the nodes' location or movement patterns and discovering their relationships leads to privacy threats. If the adversary breaks the network privacy, denial of service attacks might be performed by him. Anonymity is a critical issue in general in many wireless applications. For

example even in WLANs or cellular networks hiding the nodes' IDs, relationships and locations is important [Y.C. Hu 2005, He 2004].

Communication privacy in self-organized networks is of increasing importance for a wide range of applications to protect the location of source nodes in sensor networks and to hide the active routes and network topology in MANETs [Kamat 2005, Kong 2003]. Privacy protection for wireless networks, as described in previous chapters, can be divided into identity anonymity and location privacy, i.e. as mentioned before even in an anonymous routing protocol an attacker can still track the nodes' locations in active routes [Y.C. Hu 2005].

The problem that will be concerned in this chapter is the location anonymity of the destination of the communication. In the considered scenarios the eavesdropping adversary tries to track the route discovery messages or data flow to infer information about the destination's venue or the route established between source and destination. In existing anonymous routing protocols it is possible for highly motivated collaborative eavesdroppers to trace the RREQ (Route REQuest) and RREP (Route REPly) messages' flows to discover their origins which are supposed to be sender and the intended destination of the upcoming communication. Furthermore, the adversary can follow the data packets' flow later on to discover the active routes. The eavesdroppers can trace a message flow by observing the whole network traffic deploying a set of directional antenna or sensor nodes.

Our solution is designed to cope with the passive attacks against receiver location privacy and route privacy. We apply our ideas to ANODR [Kong 2003, Kong 2007], which is an identity anonymous routing protocol, to add destination location privacy on top of it. The main idea is to hide the route reply flow initiated from the destination among route request flow in a randomly specified area around the receiver. This is achieved by unifying RREP messages to RREQ messages since being initiated at the receiver till a random number of hops. We call this unified message type as *RDIS* (Route DIScovery message) [Taheri 2010], [Taheri]. In the following sections by RDIS-RREP and RDIS-RREQ we mean consequently the RREP and the RREQ packets modified to a RDIS packet. It is impossible for any outsider to distinguish between a RDIS-RREQ and a RDIS-RREP. After a random number of hops the RDIS-RREP will be converted to a normal RREP message by one of the nodes on the discovered route for the sake of efficiency.

Although confusing the observing adversary about the routing message flow is a key point in end point location privacy, it is not enough to hide the receiver. In fact, the adversary could follow any data packet sent over the discovered route from the source to the destination. To prevent the adversary from tracking the data packet streams toward the destination to find its location, we propose to form a ring between communicating pairs instead of a single route. The data packets that are originated by the source will traverse the whole ring. Therefore,

the receiver will be receiving the data packets without letting the adversary know which node is the final destination. We name this protocol RDIS.

Moreover, we will describe how the location of the discovered routes can be hidden from the observers using the idea suggested by the authors of the ARM routing protocol [Seys 2006] (section 2.2.3.1). They have not evaluated the performance of such an idea in [Seys 2006] but we will evaluate RDIS with and without the anonymous route idea. Including such an idea, RDIS can be used to achieve receiver location anonymity as well as route location privacy.

3.2 Related work

In section 2.2 the main existing privacy providing technologies for both wired and wireless networks were described. Among them Chaum's mixnet [Chaum 1981a] and DC-net [Chaum 1988] can be considered as the origin of many future ideas to address anonymous and private communication. As described, Mixnet investigates to remove the correlation between sources and destinations using mix nodes. A mix node is a network member that performs encryption and padding on its received messages and sends them out in a random order so that it is impossible for any outsider to distinguish which output message belongs to which input message. Chaum's mixnet was first applied to e-mail applications to provide anonymity support. Therefore, Mixnets are in fact a solution to relationship anonymity.

DC-net [Chaum 1988] is based on binary superposed sending. In DC-net the anonymity set is composed of all potential senders. Each sender shares a random secret key at least with one other user. If sender A is wishing to send a message, it should superpose the message with its exchanged secret keys. Other users superpose in the same manner (if they have no message to send they superpose zero with their shared keys). All messages are transmitted to the receiver. The sum of these messages is just the message of A, because every secret key is added twice and cancelled. Therefore as DC-net is a solution for sender privacy issue, a message can be delivered without revealing the originator.

Another example of solutions proposed for wired networks is Crowds [Reiter 1998]. Before a data request is sent to the server it is chained randomly through a number of crowds members, so that the server knows that it came from one of the members, but he has no idea about the original sender. As mentioned before, it is still unprotected against traffic analysis when an eavesdropping adversary is observing the network flows since it does not change the messages' appearance.

The protocols proposed to provide anonymity in wired networks assume having a fixed topology and usually having trusted third parties. Such solutions are not suitable for MANETs as well as any other mobile scenario in which the network topology might be changing all the time.

Most of the routing-based anonymity solutions designed for MANETs try to address the identity anonymity issue. A number of ad hoc anonymous routing protocols were described in section 2.2.3.1, such as ANODR [Kong 2003], ARM [Seys 2006], MASK [Zhang 2005] as identity anonymous routing protocols. ANODR as the first identity free routing for MANETs [Hong 2006] suggests to use route pseudonyms instead of node IDs during route discovery. Intermediate nodes use their route pseudonyms exchanged with the previous and next neighbors en route to deliver data packets. ANODR achieves ID anonymity since no node ID is revealed. In MASK [Zhang 2005] every node uses a pseudonym with its neighbors instead of its real identity every time that it moves to a new location, and every neighboring pairs share a chain of secret key and LinkID pairs. MASK explicitly puts the destination's ID in RREQ messages instead of a global trapdoor to save the processing overhead and to avoid the need of having a key agreement between every communicating pair. In RREq messages the nodes replace the pseudonym of the previous node with their own pseudonym. In the RREP phase every intermediate node will replace the LinkID field with the LinkID shared with he previous node and also will encrypt the packet with the secret key corresponding to it. The next node en route will find its pseudonym corresponding to the LinkID included in the RREP packet to find out the incoming LinkID received during the RREQ phase corresponding to that pseudonym. Then it links this pair of LinkIDs in its routing table to use them to forward data packets later on. ANODR achieves neither source/destination location privacy nor route privacy. MASK achieves destination location privacy but not destination ID anonymity.

ARM [Seys 2006] is another anonymous routing protocol for MANETs in which every two nodes should share a secret key and a pseudonym. During the RREQ phase the source creates a pair of public and private keys for the destination. The destination will use that private key to open an onion to get the link identifiers of the route to use in the data forwarding phase. ARM supposes tht the source and the destination have shared a secret key in advance which is used by the destination to open a trapdoor to be authenticated by the source as well as to be able to function as the receiver. Therefore, ARM achieves destination ID anonymity but not receiver location privacy. For route location privacy, ARM proposes to forward the RREP and data packets over some fake routes around the real discovered path to hide the route with a cloud of routes.

There are some privacy solutions proposed for sensor networks too. Kamat et al. in [Kamat 2005] proposed some ideas to support sensor networks with source location privacy. They suggest to use fake sources behaving similar to the real source to mislead the adversary over the real source location. In this solution one will need a global overview of networks to decide about the proper locations for fake sources. This paper also proposes a more effective idea called *phantom routing* in which first the message is routed from the source to a phantom source, far

from the original source and then the packet is delivered to the sink. Therefore, the phantom source leads the eavesdropping adversary who starts eavesdropping from the sink far from the real source. This protocol is not protected against a global eavesdropper.

The first work considering a global eavesdropping attacker in sensor networks is [Mehta 2007]. This protocol proposes periodic collection and source simulation. In periodic collection some nodes periodically generate and send out a dummy or real message in specified periods. Therefore, the real messages cannot be distinguished from the dummy packets. By source simulating a set of virtual objects are deployed in the field, each generating a traffic pattern similar to that of the real object to confuse the adversary about the real one. Before deployment L sensors are given different tokens. Tokens will be passed between nodes to simulate a real event. Since the movement pattern may leak the source location information, the simulated objects should have the same movement behavior like the real one which is very challenging. The privacy level depends on the number of the fake objects.

Yang et al. improved the periodic collection of above approach in [Yang 2008] by proposing a mechanism to decrease the overhead by dropping the dummy messages at some proxies in the network. They furthermore designed a mechanism to place the proxies in an efficient manner in the network.

[C 2009] is an approach to provide sink anonymity in sensor networks. This work concerns identity and location privacy of the sink node. Neither the location nor the ID of the sink is not included in data packets and therefore the adversary will not be able to find the sink information even if he can read the packets' header. The packet is sent to the sink on a number of random paths and it is possible that it never reaches the destination. Also, the sender has no idea whether the packet was received by the sink or not.

Although MANETs and sensor networks as self organized low power wireless networks have some similar properties and vulnerabilities, it is not possible to apply the sensor networks' solutions directly to MANETs. One difference is that in sensor networks usually the nodes are static and data is always sent to a powerful fixed sink node.

Some location privacy solutions for MANETS are proposed for geo-routing scenarios. For example [Wu 2008] is addressing destination location privacy for the category of ad hoc networks in which geographic information of the nodes is available. This protocol uses the location information of the destination node to generate an area including the destination to deliver the data packets to all of the nodes in that. The number of nodes inside the anonymity zone determines the privacy level provided by the protocol.

We believe that location privacy in ad hoc networks is still preliminary, as the majority of existing anonymous routing protocols concentrate on hiding the destination node ID, leaving the nodes' location identifiers unprotected. A highly

motivated eavesdropping adversary can monitor the whole or part of the network traffic and get significant information to do different denial of service attacks.

3.3 RDIS: A solution to achieve receiver location privacy in mobile ad hoc networks

In this section we will describe the main ideas proposed in RDIS. We will deploy RDIS on top of ANODR as an underlying routing protocol. However, it could be applied to some other identification-anonymous routing protocols in an appropriately designed similar way. As already mentioned, the main idea of RDIS is to cover the route reply flow among the route request flow for a number of hops from the receiver node to take the origination of RREP packets away from the receiver of the communication. For this purpose RREP packets are unified with RREQs regarding the way that they look to any observer out of the route since being initiated till a random number of hops away from the destination.

We call the unified RREP and the RREQ packets as RDIS (Route DIScovery packet type) packets. We refer to the RREP packets modified to RDIS type as RDIS-RREP packets which can only be distinguished by the nodes on the discovered route. In the following sections we describe our protocol in detail as well as the way that this idea could be applied to ANODR to add location privacy to it.

3.3.1 Attacker model

We assume the Kerckhoff's Principle about the adversary, i.e. we assume the adversary always knows every method being used in the network. The attacker wishing to break the privacy of the nodes and routes in a wireless network can range from a single machine to an omnipresent attacker roaming in the network with the ability to listen to all network traffic [Y.C. Hu 2005].

We consider two types of adversaries. The first one is a highly motivated passive eavesdropper who has the ability to monitor the traffic all over the network, for example by employing several overhearing nodes in different points of the network to cover the whole area. Our goal against this adversary is to prevent it from finding the destination's venue and also the path between communicating pairs. The second attacker considered is an internal adversary, which is a compromised node in the network. The adversary can take control of the compromised node. The private routing protocol must make it impossible for him to break the location privacy of the destination even if it is located on the route. Internal adversaries should be prevented from finding out if their neighbor nodes are source or destination of the communication even if they are on the same route. We suppose that the compromising capability of the adversary is limited.

3.3.2 Basic ideas and the contribution of RDIS

The end nodes' location privacy could be of a high importance in hostile environments. Since in most of the ad hoc routing protocols the potential global eavesdropper could be able to find the destination's location by tracking the packets initiated by him in the routing phase, it would be important to design the routing protocol such that such a risk is minimized. Our aim in RDIS protocol is to change the routing messaging such that this risk is very limited. Also, in RDIS the routes are formed as a ring instead of a single route which, as explained later, would help to achieve the receiver's location privacy.

Another aim of RDIS is to provide route privacy. By route privacy we mean to make it difficult for the global eavesdropper to discover the established route between the communicating pair. There are not many works on route privacy for ad hoc networks. One existing idea is the idea of forming a cloud of routes to hide the real route among the fake ones in ARM protocol [Seys 2006]. In ARM protocol the nodes around the discovered route broadcast the route reply and the data packets just like the nodes en route. Therefore it is not possible for the adversary to distinguish the nodes belonging to the real route. We refer to this idea of ARM protocol as *route cloud idea*.

To achieve route privacy in RDIS, we combine it with the cloud technique. One random node en route decides to change the appearance of the RREP packet from RDIS-RREP to a RREP. We call that node as the *suspected node*. From the *suspected node* on, we use the *cloud idea* of ARM to route the RREP message toward the source node. So the adversary will not be able to distinguish the discovered route.

RDIS is designed on the base of message type unification idea in the routing phase. As mentioned before, an adversary is able to identify the destination node's venue during the route discovery phase by looking for nodes sending out a RREP message after broadcasting a RREQ message. To overcome this problem we try to make it impossible for the eavesdropper to distinguish between RREQ and RREP messages. RREQ and RREP messages differ in message type, size and packet fields. The size difference is equalized by adding an additional field to the RREQ messages containing random bytes which are changed at every node. The pattern in which the packet fields are updated could be a clue to the adversary to distinguish the type of a packet even if he is not aware of the packet type. For example the adversary may know that a specified field in the route request packets is the route request identifier which does not change at different hops while the same field is a hop-by-hop encrypted information in the route reply packets of the same protocol. Therefore he will be able to distinguish between RREQ and RREP packets only based on this difference.

In RDIS the route reply packet flow will seem to be the continuous of the route request flow. We design this protocol such that a RDIS-RREP packet cannot be distinguished by the nodes not belonging to the discovered route. Since

the nodes locating on the discovered route still need to be able to differentiate between RDIS-RREQ and RDIS-RREP, the RDIS-RREP messages require another possibility to be identified. Normally in ad hoc routing protocols the nodes en route use the keys or secrets generated in the route request phase to check if a received route reply packet is intended for them. In RDIS the nodes out of the route cannot even distinguish between a RDIS-RREP packet and a RDIS-RREQ one and only the nodes en route can distinguish the RDIS-RREP ones using their keys.

The *suspected node* who is selected randomly will modify the RDIS-RREP packet to a normal reply packet. So even a global eavesdropper cannot see which node is the real origin of the reply packet. When a node out of the discovered route hears a RDIS-RREP message it would behave exactly as it does about a RDIS-RREQ message if the TTL field in the packet is larger than zero. As shown in Figure 3.1 after a random number of hops the reply packet is changed to a normal RREP packet by one of the intermediate nodes. We do not continue forwarding the reply packet in RDIS format toward the source node because that will cause a high overhead due to two reasons. First, the RDIS-RREP packet will be broadcast by every node receiving that till $TTL = 0$, and second the size of a RDIS-RREP packet is larger than a normal RREP packet and regarding routing overhead it leads to a more efficient point.

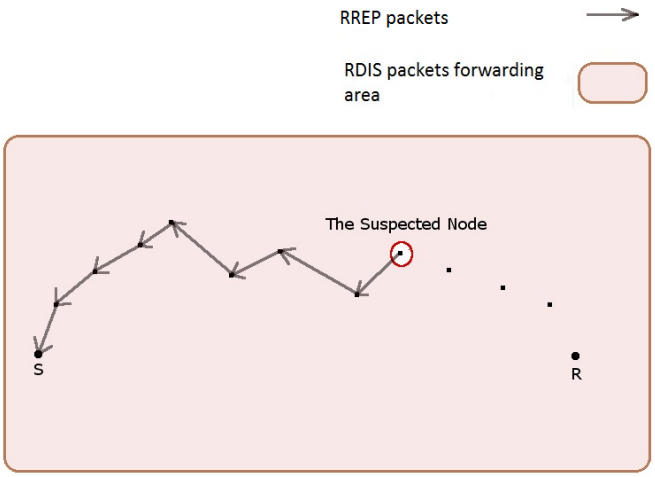

Figure 3.1: The RREP packets are hidden among RDIS flow from R to the suspected node

After hiding the route reply flow in the route request signalling, another problem remains to solve. Adversaries may also follow the data packets along the route to identify the end nodes of the communication. The data packets will stop at the destination and the adversary will be able to find the destination

following the data flow. To solve this problem we circulate any data packets over a ring path which includes the source and the destination. We form two routes from the source to the destination and build a ring of routes as described later. The source node will send all its packets on the route discovered first (route1). The destination node will forward the packets on the second discovered route (route2) back to the source node (See Figure 3.3) and the packet will be stopped there.

3.3.3 Deploying RDIS on top of ANODR

An overview of ANODR protocol is given in section 2.2.3.1. In this section we are going to describe how the ideas of RDIS can be applied to ANODR to provide destination location privacy as well as route privacy. To apply RDIS to ANODR we need to unify the appearance of the route request and the route reply messages such that the RDIS-RREP flow seems to be part of the RDIS-RREQ flow to any outsider without losing the routing functionalities.

For this purpose several properties should be considered. One is properly unify the size of RDIS-RREP and RDIS-RREQ packets to prevent the outsider from distinguishing them. Another is that the appearance difference from the RDIS-RREQ packet the destination node receives and the RDIS-RREP packet it initiates should be similar to the difference between a received RDIS-RREQ packet received at any other node and the RDIS-RREQ packet broadcast consequently by it. Therefore the initiation of the RDIS-RREP message would look like a part of the RDIS-RREQ flow. Also every field of one of these two message types should change with the same pattern as the other one. For example, the sequence number which is a fixed field in RDIS-RREQ should be preserved the same in the corresponding RDIS-RREP flow. As a matter of course we change the content of the message type field in both of them to the same packet type, $RDIS$. When a node receives a $RDIS$ packet with a new $seq\#$, it will generate a random number between 0 and 1. If the number is less than a fixed parameter P_f the node will proceed with the packet, otherwise it will do nothing and therefore discard the packet. If the node decides to proceed with the received packet it will record the $seq\#$ in its routing table and will proceed with the message to follow the ordinary ANODR behavior (described in section 2.2.3.1). When the destination node receives the RDIS-RREQ message it generates the corresponding RDIS-RREP packet. It decreases the received TTL by one. The RDIS-RREP packet includes a sequence number field filled with the same $seq\#$ of the corresponding RDIS-RREQ (in regular ANODR there is no sequence number or TTL in reply packets).

The global trapdoor is preserved in RDIS-RREP although it will not be used on the reverse path from the receiver to the source. We change $\{K_{seed}\}_{PK-1time}$ to $\{REPLY, K_{seed}\}_{PK-1time}$ in the RDIS-RREP packet. In order to match the size of the RDIS packets we need to add an additional field in the RDIS-RREQ

packets filled with random data. So all in all a RDIS-RREQ packet will look like

$$< RDIS, TTL, seq\#, global\ trap, onion, PK-1time, random\ field >$$

and a RDIS-RREP packet will look like

$$< RDIS, TTL, seq\#, global \\ trap, \{REPLY, K_{seed}\}_{PK-1time}, f_{K_{seed}}(Proof_{des}, onion) >$$

The adversary may distinguish between the RDIS-RREQ and RDIS-RREP messages because he knows that the onion length in RREQ messages increases as the message nears the destination and the onion length in RREP messages decreases as the message traverses further from the destination. Therefore the onion length should be fixed. In an improved version of ANODR the length of the onion is fixed at 128 bit [Kong 2004]. Every node applies its symmetric key encryption on the 128 bit long onion. In RDIS, we use this mechanism to prevent the adversary from using the varying length of the onion to analyse the message type or the distance from the destination.

When a node receives a $RDIS$ message while it has forwarded another $RDIS$ message with the same $seq\#$ before, it will try to open $\{REPLY, K_{seed}\}_{PK-1time}$ using its one time public key generated during the RREQ phase. If after such a decryption the node can see the REPLY tag it realizes that this packet is a RDIS-RREP intended to it. Then it will generate a random number between 0 and 1. If this number is greater than a fixed parameter P_{modify} it will decrease TTL by one and replace the K_{seed} and the onion with its own (see section 2.2.3.1). Otherwise, it will change the RDIS-RREP message to a normal RREP message as shown below, but the TTL field will be preserved to be used for the route cloud idea. So one of the nodes en route randomly will change the RDIS-RREP packet to a normal RREP as follows, which except having the TTL field is the ordinary reply packet format in ANODR:

$$< RREP, TTL, \{K_{seed}\}_{PK-1time}, f_{K_{seed}}(Proof_{des}, onion) >$$

Let's assume T_{rep} is the maximum time that a source node waits to receive the corresponding RREP after initiating the RREQ. We consider the recorded one time public keys at the nodes as fresh keys during T_{rep} seconds after being generated. When a node receives a packet like the above RREP packet and it has a fresh one time public key it will use it to find out if the packet is intended to it (by opening the onion as in ordinary ANODR). If so, the node will modify the reply packet as described in 2.2.3.1 and will also decrease TTL by a random number among 1,2,3 and 4. Therefore this packet will be forwarded on the discovered route normally till it reaches the destination. When a node that is not located on the discovered route receives such a packet and it realizes that the packet is not intended to it, it will generate a random number among 1,2,3

and 4 and will decrease the TTL by that. It will also replace the next two fields with random bits without changing the packet size and broadcasts the packet. Therefore a cloud of routes will be formed around the route and the discovered route will be hidden among them. This will provide the protocol with route location privacy because the nodes out of the route behave just like the nodes on the route regarding route reply packet forwarding.

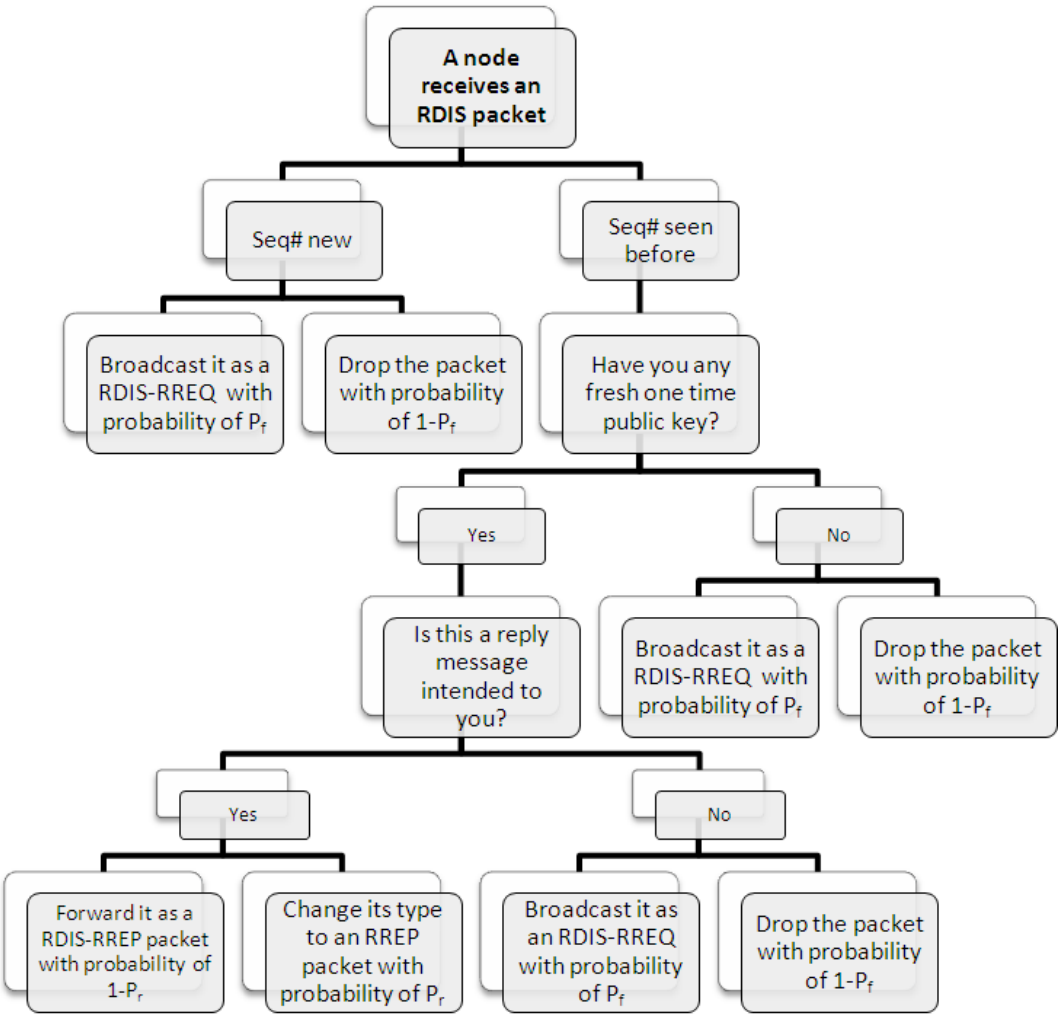

Figure 3.2: RDIS Packet flow

3.3.3.1 Ring route idea

As mentioned before, in RDIS instead of a single route between the source and destination, a ring route is formed between the two communication end nodes such that they are both located on it as shown in Figure 3.3. As the RDIS-RREQ

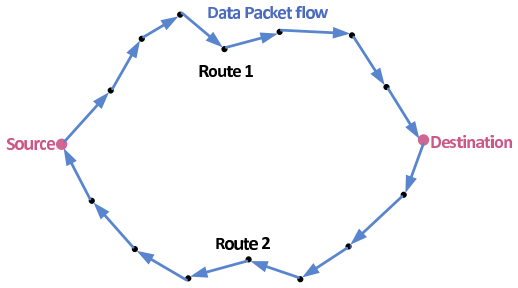

Figure 3.3: The ring route idea

message is broadcast in the network the receiver node might receive the route request of the same source node for several times. In RDIS the receiver node responses to the first two received RREQ messages, and therefore two routes will be formed between the communication end nodes. This two routes can form a ring such that the source and the destination are both locating on it.

The established routes are used bidirectionally. It is possible because every two neighboring nodes on a route are sharing a link pseudonym pair which are used to forward the data packets over the route. When the destination receives any data packet it forwards it to the first node on the other route and the data packet will be forwarded (in the reverse direction) through that route to reach the source. Then the source node will discard it. Therefore an eavesdropper is unable to identify precisely the destination node, as it could be any node on the ring of routes.

As mentioned before, in RDIS all nodes act with a certain probability when establishing the routes, i.e. every received RREQ packets are proceeded by every node with the probability of P_f (which means some nodes will decide not to broadcast the RREQ). One consequence of this property is that the first discovered route is not necessarily the shortest one and also the first two discovered routes might be quite far from each other (because the intermediate nodes are chosen quite randomly and the two paths are not necessarily the shortest ones). When the source node realizes that two routes are discovered it starts sending data packets to the receiver through the first one.

Figure 3.2 shows the overall route discovery packet flow of RDIS.

3.4 Privacy Analysis

Normally, privacy level cannot be measured in network simulations but an appropriate mathematical model can help to calculate such a metric. In this section we build a mathematical model to evaluate the privacy provided by RDIS. What we measure here is the adversary's uncertainty level about the privacy information of the network. We will study the effect of the P_{modify} parameter, the number of nodes captured by the adversary (internal adversaries) and the network node

density on the achieved privacy.

We use Shannon entropy for privacy calculations. For a given discrete random variable X with the probability distribution function of $P(X)$ the Shannon Entropy is defined as

$$H(X) = -\sum_{i=1}^{N_X} P(x_i) log_2(P(x_i)) \tag{3.1}$$

The maximum value of the entropy for any random variable is when every events can happen with the same probability. So if for $0 \leq i \leq N_X$ we have $P(X_i) = \frac{1}{N_X}$ the entropy will have its maximum value of $log_2(N_X)$.

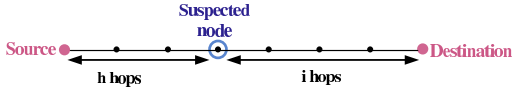

Figure 3.4: Network model in one dimension

In the following calculations we consider to have a uniform node distribution in the network with the node density of about $100 nodes/km^2$ and the number of nodes located k hops away from a node is supposed to be $c_k = 8k$. Also, the average length of a route is $L = 15$. The adversary knows that not the *suspected node* itself but another node with a random distance to it is the real destination.

The privacy level represents the adversary's uncertainty about the destination's location. The adversary eavesdrops in the whole network to detect the point of initiation of a RREP packet to find the communication's destination. Of course in RDIS this point is not the destination's location, but the adversary can use his knowledge about the protocol to guess with some uncertainty where the destination could be located. We suppose that the adversary is a global eavesdropper. Weaker eavesdroppers will have less chances to guess the destination's location. In some of the following parts we will consider the adversary with node capturing abilities too.

We first describe our model in a single dimensional network. Suppose that the network is established as Figure 3.4. D and S are the receiver and the source and A is the node changing the reply packet from RDIS-RREP to RREP, i.e. the *suspected node*. Let l be the distance from the destination to the node converting the reply packet type, the *suspected node*.

According to RDIS design the probability that the reply packet type is changed after i hops from the receiver would be

$$P(l(D, A) = i) = P_{modify} \times (1 - P_{modify})^i : 0 \leq i \leq L - 1 \tag{3.2}$$

Actually $i = 0$ means that the destination itself decides to perform the reply change. Obviously we need to have $\sum_{i=0}^{L} P(l(D, A) = i) = 1$. In reality in RDIS

the packet type change never happens on hop number L (at S). So we need a proper definition for $P(l(D,A)=L)$ which should also satisfy

$$\sum_{i=0}^{L-1} P(l(D,A)=i) + P(l(D,A)=L) = 1 \tag{3.3}$$

For the following power series we know that the following equation holds.

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \tag{3.4}$$

So we have

$$\sum_{i=0}^{\infty} P(l(D,A)=i) = \sum_{i=0}^{\infty} P_{modify} \times (1-P_{modify})^i = 1 \tag{3.5}$$

We define

$$\begin{aligned} P(l(D,A)=L) &= \sum_{i=0}^{\infty} P(l(D,A)=i) - \sum_{i=0}^{L-1} P(l(D,A)=i) \\ &= 1 - \sum_{i=0}^{L-1} P(l(D,A)=i) \end{aligned} \tag{3.6}$$

$P(l(D,A)=L)$ will actually model the case that the packet reaches the S as a RDIS-RREP packet. If the adversary observes that the RREP packet is initiated at node A, then the probability that the node locating i hops away from A be the destination is $P_{modify} \times (1-P_{modify})^i$. So with this observation, his uncertainty about the destination location is as follows.

$$\begin{aligned} H = &- \sum_{i=0}^{L-1} [P_{modify}(1-P_{modify})^i][log_2(P_{modify}(1-P_{modify})^i)] \\ &- [1 - \sum_{i=0}^{L-1} P(l(D,A)=i)][log_2(1 - \sum_{i=0}^{L-1} P(l(D,A)=i))] \end{aligned} \tag{3.7}$$

We study how the P_{modify} parameter, the internal adversaries and the network node density affect the achieved privacy. Figure 3.5 shows the effect of changing P_{modify} on the expected value of l for $L=8$ and $L=5$. The higher the expected value of l (the lower the P_{modify}) the higher the privacy level (it will be more difficult for the adversary to find the real destination as the *suspected node* is further). So the value of P_{modify} can be chosen depending on the required privacy level.

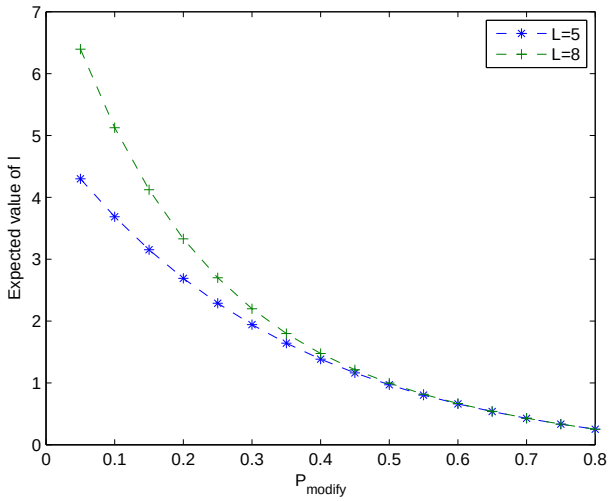

Figure 3.5: Expected Value of Destination's Distance from the Suspected Node

In the reality we need to consider a two dimension network, i.e. we have more than one node in every hop distance from the suspected node let's say N_i nodes in the hop distance i, then the probability that any given node locating i hops away from A is the destination would be $P(l(D,A)=i)/N_i$. In this case the entropy of the random variable of receiver's distance from the adversary is as follows.

$$\begin{aligned} H = & -\sum_{i=0}^{L-1} N_i \frac{P_{modify}(1-P_{modify})^i}{N_i}[log_2(\frac{P_{modify}(1-P_{modify})^i}{N_i})] \\ & - N_L \frac{P(l(D,A)=L)}{N_L}[log_2(\frac{P(l(D,A)=L)}{N_L})] \\ & = -\sum_{i=0}^{L-1} P_{modify}(1-P_{modify})^i[log_2(\frac{P_{modify}(1-P_{modify})^i}{N_i})] \\ & - (1-\sum_{i=0}^{L-1} P(l(D,A)=i))[log_2(\frac{1-\sum_{i=0}^{L-1} P(l(D,A)=i)}{N_L})] \end{aligned} \tag{3.8}$$

Another clue that the global adversary could get is to follow the detected RREP packet to see how many hops the rest of the route from the *suspected node* to the source node is consisting of. Let h be the length of the route from the *suspected node* to the source node (see Figure 3.4). Since he knows the average route length he has more information about the possible distance of the suspected node to the real destination. Figure 3.6 shows the anonymity set size for $L=5$ and $L=8$. As seen in the graph when h is larger, i.e. when the adversary knows that a longer part of the route is not hidden, then he knows that

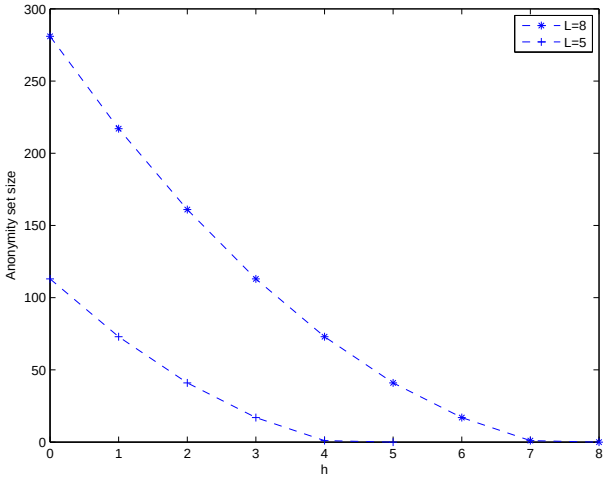

Figure 3.6: Anonymity Set Size for different h values

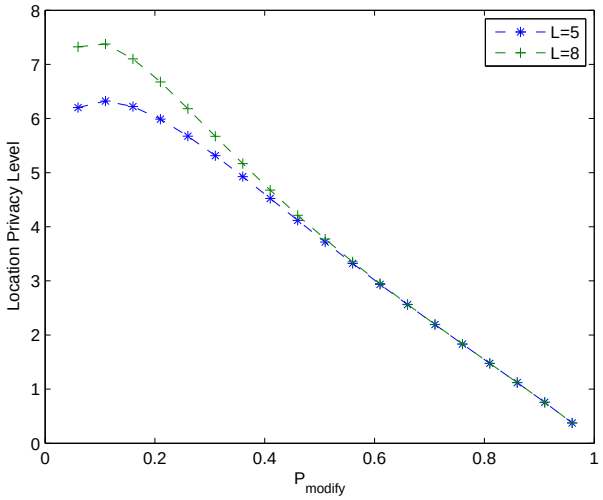

Figure 3.7: Location Privacy Level changing with P_{modify}

the destination is not far from him and the anonymity set size gets smaller and the attacker has less uncertainty about the location of the destination. Figure 3.7 shows the average of privacy level over h for different values of P_{modify} for $L = 8$ and $L = 5$. As expected the privacy level for longer routes is higher.

If the adversary has node capture abilities then the privacy level will decrease since the captured nodes are out of the anonymity set from his point of view (since the adversary is willing to remain undetected he does not perform any active attack). Figure 3.8 shows the average location privacy when some nodes

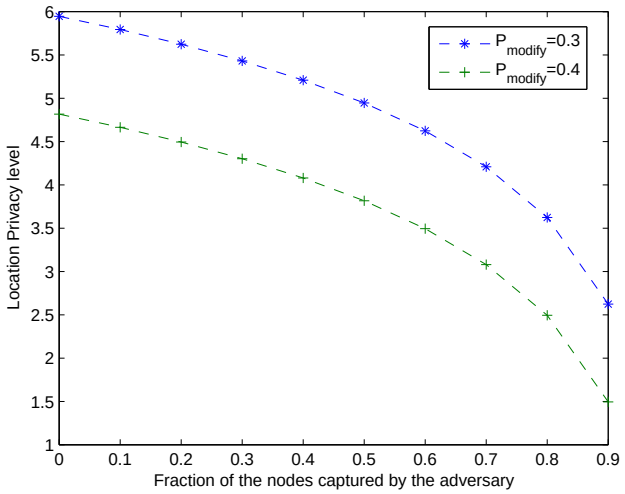

Figure 3.8: Privacy level for different captured nodes fraction averaged over h

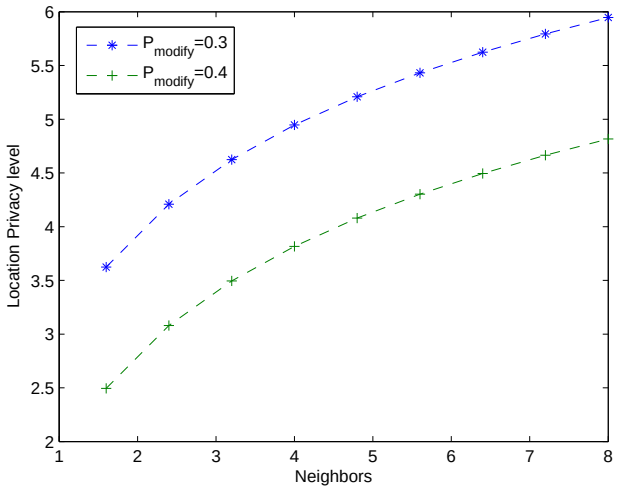

Figure 3.9: Privacy level for different node densities averaged over h

are captured for $P_{modify} = 0.3$ and $P_{modify} = 0.4$. We also study the impact of node density on the privacy. With lower node density it is easier for the adversary to guess about the location of the receiver among the existing nodes. Figure 3.9 shows the impact of the node density in terms of number of neighbors per node on the destination privacy level. Note that the corresponding node density to 8 neighbors per node is about 100 $nodes/km^2$. This figure also shows the privacy parameter for $P_{modify} = 0.3$ and $P_{modify} = 0.4$.

3.5 Performance evaluation

In this section we use QualNet [Qua] to implement RDIS protocol to see in which way the routing performance is affected by the location privacy ideas of RDIS. QualNet is a packet level simulator for wired and wireless networks. We compare the performance results of RDIS to ANODR as the underlying routing protocol. Obviously since we did not change ANODR's basic routing structure but added destination location privacy to it we do not expect to have an improved routing performance, but we expect RDIS performance metrics to be acceptable as a routing protocol providing location privacy.

3.5.1 Simulation model

Distributed Coordination Function (DCF) of IEEE 802.11 is used as MAC layer. We have simulated an ad hoc network with 50 nodes initially uniformly distributed in a 1500×300 m^2 field. The simulation time is 15 minutes. The transmission range of nodes is 250m and the node mobility model is *random way point model* with pause times of 30 seconds. At application layer constant bit rate (CBR) sessions with data packet size of 512 bytes generated at a rate of 4 packets/second are modelling the communications. During each simulation a constant number of concurrent short lived CBRs are maintained. In order to simulate more realistic traffic the length of a single CBR session varies between 120 and 240 seconds. The evaluation metrics include: *Delivery fraction ratio-* The ratio of the data packets delivered to the destinations to those sent by the sources. *Average data packet end-to-end delay-* The average time difference between generation of data packets and their reception at the destination, including buffering time due to route discovery latency, processing delays at other layers, queuing at the interface queue, retransmission delays at the MAC layer, propagation and transfer times. *Normalized control bytes-* Number of the routing packets transmitted per one successfully delivered data packet.

3.5.2 Simulation results

Figure 3.10 shows the data delivery ratio when the node speed is increased from $0m/s$ to $10m/s$ using random way point model. To have a more reliable result we run the simulation 5 times for randomly chosen communication pairs and show the average as the final result. In each simulation run 5 concurrent short lived CBRs are maintained all the time. The values of network parameters are chosen as $P_{modify} = 0.4$ and $P_f = 0.75$. As the nodes' speed increases the difference between the delivery fractions of RDIS and ANODR increases slightly, but even in high mobilities the delivery fraction of RDIS is higher than 0.93. This results are showing that the privacy properties added to the protocol does not affect the data delivery performance noticeably.

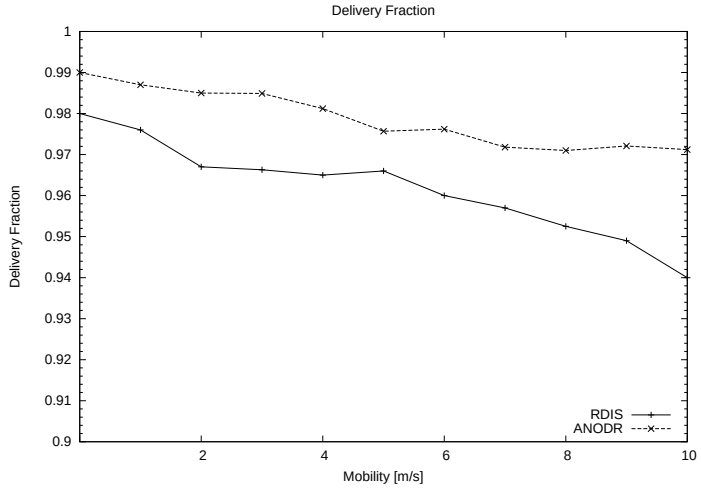

Figure 3.10: Data delivery fraction for different node mobilities

It would be interesting to see how much routing overhead is added to the protocol due to the privacy supporting ideas of RDIS. We expect a higher control byte overhead in RDIS compared to ANODR due to discovering and maintaining the second route in the ring routes. Also in RDIS-RREP packet propagation phase the packets need to be broadcast by some extra nodes to look like a part of RDIS-RREQ flow. Figure 3.12 shows the normalized control bytes of RDIS compared to ANODR for different node mobilities. Obviously P_{modify} affects the amount of control byte overhead in RDIS as with higher P_{modify} values the reply packets would be converted to the ordinary RREP ones earlier. Figure 3.11 shows the normalized control bytes with growing P_{modify}. In this simulation the number of concurrent CBRs is 5 and nodes' speed of the *random way point mobility model* is fixed at $2m/s$. As the graph is suggesting, for higher values of P_{modify} the routing overhead is less. The reason is that RDIS-RREP packets are larger than the ordinary RREPs in size and are also broadcast as RDIS packets in the network. So the sooner the reply flow changes to RREP packets (higher P_{modify} value) less overhead is expected. Comparing Figure 3.11 and Figure 3.7 one can see the trade-off between the privacy and the overhead.

Figure 3.13 presents the average end-to-end data packet delay in RDIS and ANODR. As expected the delivery delay in RDIS is a bit higher, since the source node needs to wait for the second route of the ring route to be discovered before it starts transmitting data packets. Therefore the data packets generated before the second route is discovered will experience a higher latency which will affect the average delivery delay to some extent. Even in highest speeds the delivery delay of RDIS is less than 60ms.

In order to investigate the effect of traffic load on the performance metrics of RDIS we performed another set of simulations. In the following simulations

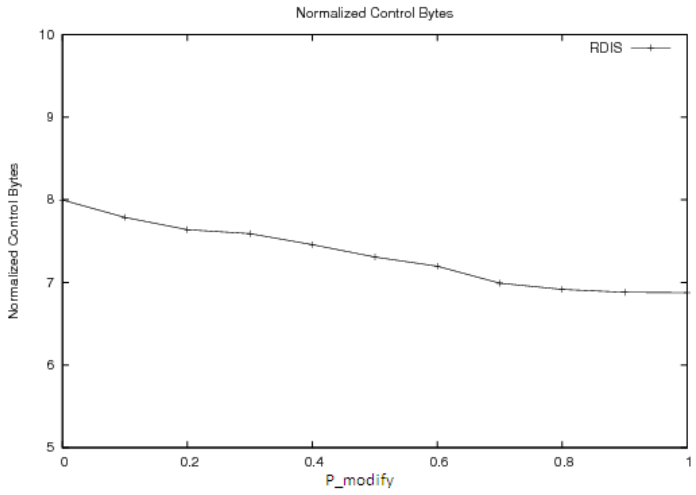

Figure 3.11: Normalized control bytes for different P_{modify} values

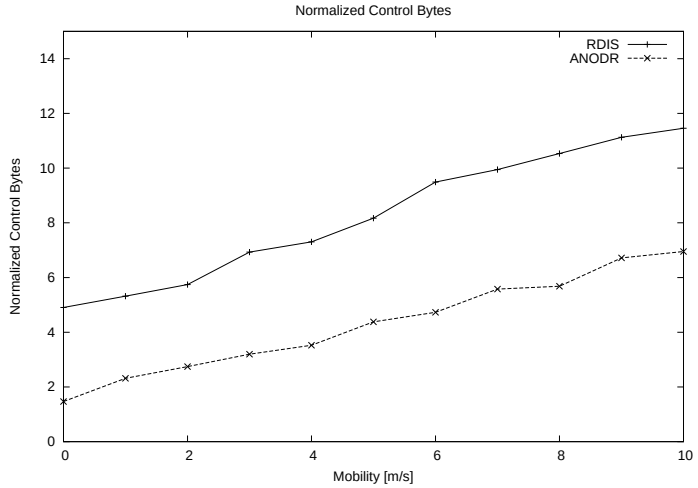

Figure 3.12: Normalized control bytes for different node mobilities

5 simulation runs are done. In this simulation series the number of concurrent CBRs is increased from 5 to 25. The packet delivery fraction is shown in Figure 3.14. Figures 3.15 and 3.16 present the end-to-end delivery delay and the normalized control bytes for different traffic loads. It can be seen that compared to ANODR the performance of RDIS in high traffic loads is still satisfying.

The last simulation series in this chapter are about the performance parameters of RDIS when the cloud idea is implemented too. As described before in cloud idea in order to confuse the adversary about the real route between the source and the destination the packets are forwarded in a cloud of fake routes.

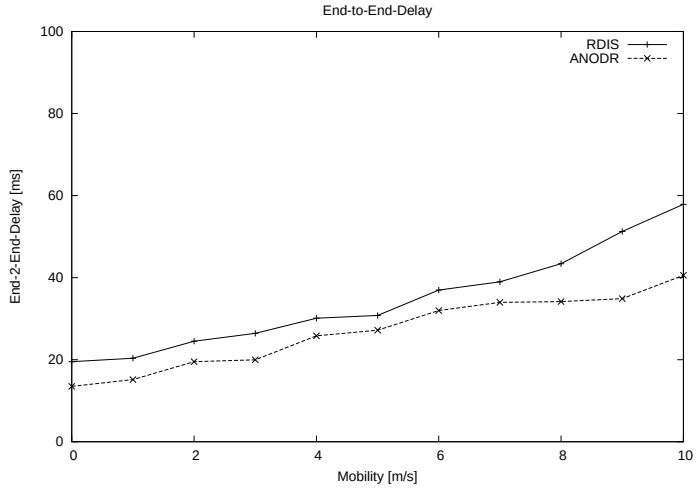

Figure 3.13: Average data packet end-to-end delay for different node mobilities

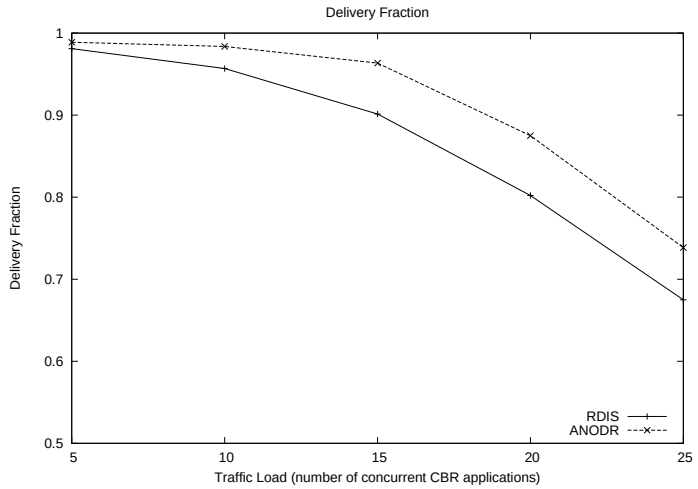

Figure 3.14: Data delivery fraction for different traffic loads

This clearly will decrease the routing performance compared to the RDIS without cloud idea. In this set of simulations, we maintain 3 concurrent CBR applications during the simulation time. Figure 3.17 shows the delivery ratio and Figures 3.19 and 3.18 show the average end-to-end delivery delay and normalized control traffic amount. It can be seen that the data delivery ratio is just a little bit lower compared to ANODR, but with cloud routing the overhead and delay are more affected. This is because forwarding the route reply packets on the cloud of routes means an extra routing overhead and this in turn will cause more congestion of the nodes. Therefore, the data packets may get lost at a higher rate and they will experience a higher end-to-end delay because of being retransmitted.

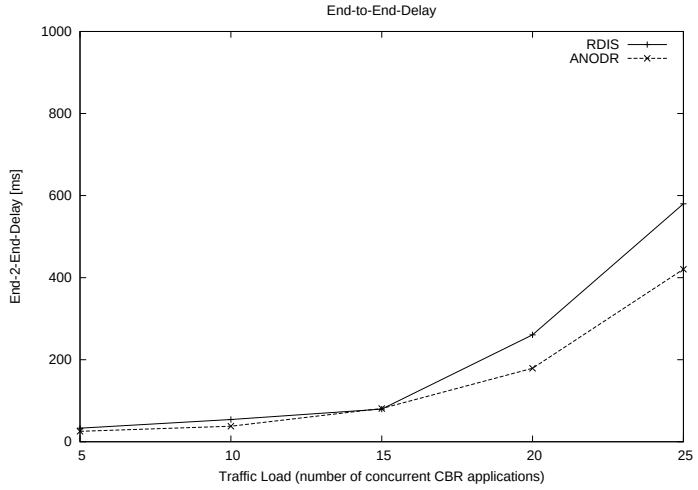

Figure 3.15: Average data packet end-to-end delay for different traffic loads

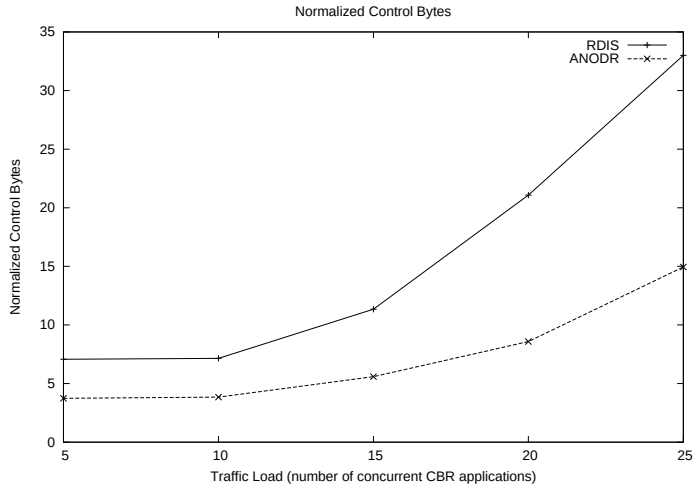

Figure 3.16: Normalized control bytes for different traffic loads

3.6 summary

In this chapter RDIS as a location privacy routing protocol for MANETs was proposed which is based on some modifications in the packet flows in the network layer. We applied RDIS ideas on top of ANODR as the ID anonymous routing protocol while it could be applied to some other anonymous MANET routing protocols in the appropriate way.

The main idea here is to hide the real destination among an anonymity set of nodes in the network by mixing the route request and route reply packet flows. The privacy level depends on the protocol parameters as well as the node density. The achieved level of location privacy is valid even against a global

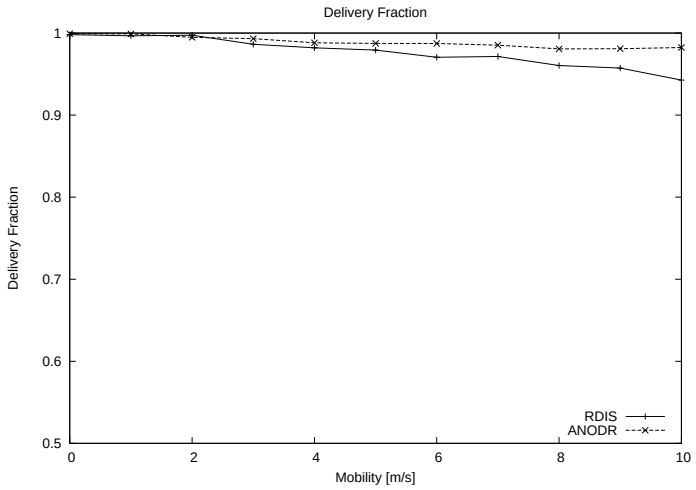

Figure 3.17: Data delivery fraction when cloud idea is added to RDIS

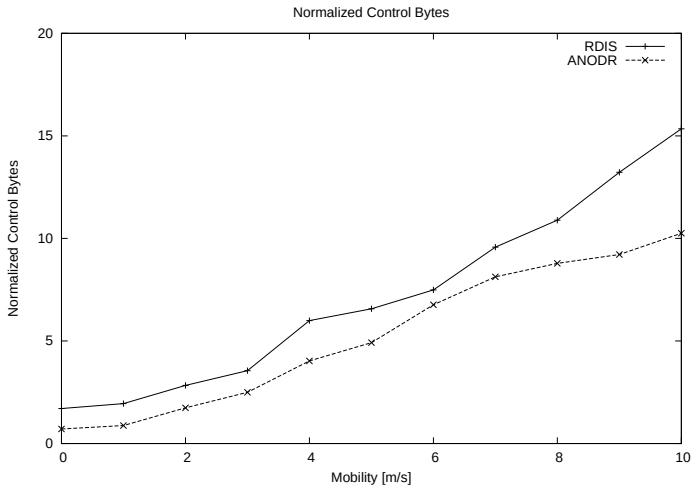

Figure 3.18: Normalized control bytes when cloud idea is added to RDIS

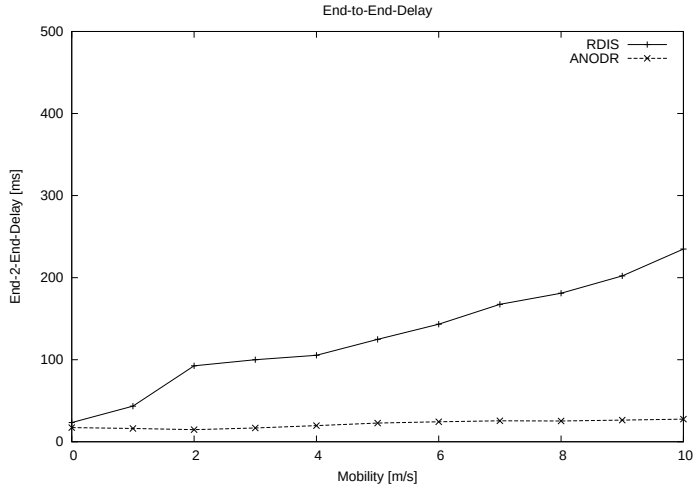

Figure 3.19: Average end-to-end delay when cloud idea is added to RDIS

eavesdropping adversary and decreases when the network includes some internal adversaries which are modelled as captured nodes.

We evaluated the privacy level of RDIS by introducing a mathematical model for the adversary's uncertainty about the destination's location. In this evaluation we showed that RDIS provides a good level of location privacy which depends on some network parameters.

We also implemented the protocol to study the performance of RDIS such as packet delivery ratio and the control packets' overhead. We compared the performance metrics of RDIS to the same metrics of ANODR to see the trade off between privacy and performance efficiency. Our simulation results show that RDIS is satisfying destination privacy requirements to a high level while still being efficient. If route anonymity is also required, RDIS with cloud routes can be used to hide the routes' position.

Chapter 4

Privacy in multicast ad hoc routing

Although operating as groups is required by many ad hoc applications, there is only a sparse work on anonymous multicast routing algorithms in the literature. In this chapter we propose an *Ano*nymous *Mul*ticast routing protocol for ad hoc networks called AnoMul [Taheri 2009]. In AnoMul we extend the idea of identification free route discovery proposed by Jiejun Kong et al. in [Kong 2003] to multicast routing and also provide some privacy mechanisms to conceal the privacy information of the multicast communication from the outsiders.

4.1 Introduction

There are variety of existing and emerging ad hoc applications with high importance of *multicast* where a single member needs to transmit data or voice/video packets to other group members. Video conferencing , multimedia streaming, MANET auto-configuration, emergency warning in VANET systems or multiparty gaming in a conference room are examples of multicast applications of ad hoc networks. When multicast is used instead of multiple unicast the overhead is reduced and multicast receivers can receive data while their individual addresses are unknown or changeable to the sender [Kunz 2004].

Security and anonymity are critical requirements that could be demanded by group based communications just as in unicast scenarios. A customer deploying MANETs for above mentioned applications could demand privacy of the group members and transmission contents as a property of the network. Furthermore, in security critical applications of MANETs such as military data transmission or law enforcement scenarios anonymity issues would be much more important. A typical example of anonymous multicast communication set-up could be in a battlefield where the commander is sending data about the mission to the soldiers or in a video conference where a user needs to send data or communicate to another (or a subgroup of) user(s) without revealing it to everybody.

Efficient security and privacy are more complicated issues in multicast routing compared to the unicast scenarios. Group key management and group membership policies are special issues to multicast protocols. In multicast the sender needs to deal with the privacy of its connections to a group of receivers.

Anonymity and the location privacy of the group leader is another issue in multicast communication.

We believe that multicast anonymity is one of the noticeable current challenges of MANETS, while among the proposed anonymous algorithms for ad hoc networks there is only a very little number of works focusing on group based applications. In this work we introduce an anonymous multicast routing protocol for MANETs, called AnoMul. AnoMul is a receiver initiated mesh-based multicast protocol, i.e. if a group member needs to receive the group data it should connect to the mesh first. The receivers do not need to send their join requests to the group leader, as it is enough to contact another group member which is already part of the mesh. This property makes AnoMul simple and fast. We apply the idea of packet type hiding to provide location privacy to the group senders and the group leader in multicast communication. This mechanism prevents potential eavesdroppers from noticing the ongoing group activities.

4.2 Related work

4.2.1 Multicast routing protocols

There are two types of multicast communication structure in mobile ad hoc networks: tree-based multicast and mesh-based multicast protocols. Although tree-based protocols are more efficient in terms of routing overhead and scalability (since only a tree is established to deliver multicast packets to the receivers), they are not stable enough in mobile ad hoc environments where the network topology is very dynamic. There is only one route from the source to each destination and some routes are broken due to the dynamic nature of the network, so the delivery ratio of the group could be highly affected in high mobility scenarios. MAODV [Royer 1999] is a tree-based multicast protocol for MANETS.

Mesh-based protocols provide higher redundancy to deliver data packets to the receivers through multiple possible paths and show a better functionality in high mobility situations. CAMP [Garcia-Luna-Aceves 999] and ODMRP [ju Lee 1999] are examples of mesh-based multicast protocols. Furthermore, there are hybrid multicast routing protocols too, which combine tree and mesh-based approaches together to achieve an efficient and robust multicast communication. PUMA [Vaishampayan 2004] and AmRoute [Xie 2002] are examples of hybrid multicast protocols.

Table 4.1 shows the MANET multicast structure classification and mentions some examples for each category.

Many of proposed multicast routing protocols for MANETs depend on their underlying unicast protocol, for example MAODV , PUMA , CAMP and AmRoute.

Table 4.1: Multicast routing protocols in MANETs

Multicast topology	**Examples**
Tree-based	MAODV, AMRIS, MZRP, MOLSR
mesh-based	ODMRP, CAMP, Patch ODMRP
Hybrid	AmRoute, PUMA, DMZ, MCEDAR, FGMP

In this section we review the main properties of few multicast ad hoc protocols, namely MAODV, ODMRP and AmRoute.

MAODV (Multicast AODV)

MAODV is a tree-based multicast approach designed as an extension to AODV to add multicast functionalities to it. The join request and replies are the same as RREQ and RREP messages in unicast AODV.

When a node decides to join the multicast tree, it broadcasts a join RREQ message and only the nodes who are already a member of the tree will respond to it. The joining node selects the route with the biggest sequence number with shortest hop count to the tree. Then, an activation message is forwarded along the selected route hop by hop to enable the entry for the source node in the multicast routing tables of the nodes en route. The first node joining the tree would be the group leader who will be responsible to maintain the tree by sending Hello messages. When a node leaves the group, the tree will be repaired locally which is the responsibility of the first node on the broken link.

ODMRP (On-Demand Multicast Routing Protocol)

ODMRP is an on demand mesh based multicast protocol initiated by the source node. When a source node has data packets to send to the group and has no routes to the receivers it sends a Join Data packet. Upon receiving the Join Data the intermediate nodes store the ID of the upstream node and re broadcast the packet. The receivers will respond to the Join Data by sending a Join Table to form a forwarding group between the source and the receivers. On receiving a Join Table, a node checks if its own ID matches one of the next node IDs' of any of the entries in that. If so, the node supposes itself as one of the nodes on the route to the source node and therefore broadcasts Join Table after updating it. Finally the Join Table will reach the multicast source through the shortest path.

The mesh is updated by the source node by sending periodic Join Data to the mesh. If a multicast source decides to leave the group since it has no more data packets to send, it just stops sending Join Data messages. The receivers also do not send leave messages when quitting the group, they just stop replying to the source node. ODMRP can also be used as an efficient unicast data delivery routing protocol. Figure 4.1 shows the join reply signalling from the group

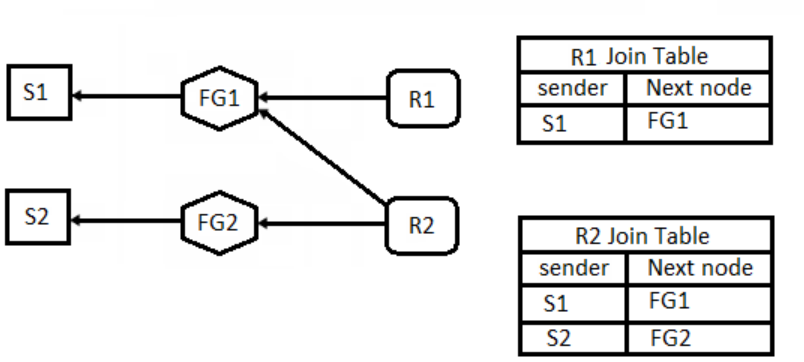

Figure 4.1: ODMRP: the Join Table messages

receivers to the sources.

According to several studies, e.g. [Md. Saiful Azad 2009], [Sung-Ju Lee 2000] and [M 2011], ODMRP is a high performance multicast prtocol compared to many existing approaches.

AmRoute (Ad hoc Multicast Routing)

AmRoute is a hybrid multicast protocol based on a shared tree. Initially, every group member declares itself to be a core of a mesh of size one (itself). Each core periodically floods Join REQS packets. When a group member receives a Join REQS from a core it replies with a Join ACK. Then this two nodes record each other as mesh neighbors. Actually, when a core receives a Join REQS from another core it replies and a bidirectional path is formed between them. Then one of the two cores will be the new core.

The user-multicast tree is composed of group members only. Each group at least has one core who is responsible for finding new members and forming the tree and maintaining it. AmRoute assumes to use an underlying unicast communication protocol to establish unicast routes between neighboring nodes. Given that the unicast connectivity between members is maintained, the tree will be robust to network changes. According to the network connectivities the core can migrate dynamically in the mesh. Robustness is a more important goal in AmRoute than bandwidth and latency.

The tree will be created once the mesh is formed. Each core periodically sends out Tree-Create packets to its mesh neighbors. If one of its mesh neighbors

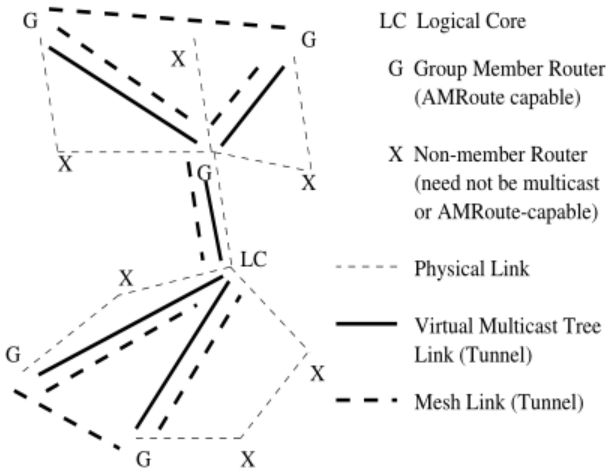

Figure 4.2: AmRoute: a user-multicast tree

receives such a packet it will forward it to all other mesh links. A duplicate Tree-Create will be responded by a Tree-Create-NAK along the path. The node receiving a Tree-Create-NAK considers that link as a mesh link not a tree link. When a node wants to leave the group, it will send a Join-NAK to the neighbors and will stop forwarding data packets. Figure 4.2 shows a user-multicast tree which connects 6 members.

AmRoute uses mesh connections as virtual links to establish the tree, therefore when the topology changes the tree do not need to be readjusted as long as the mesh routes are available.

4.2.2 Multicast anonymous protocols

Anonymous unicast routing in MANETs has been a popular field of research in the past decade and lots of different approaches, e.g. [Kong 2003], [Zhu 2004], [Zhang 2005], [Taheri 2010], [El-Khatib 2003], have been suggested. On the other hand, the efforts done in security enhancing in MANET multicast routing mainly focused on node authentication and group key exchange algorithms instead of secure routing and specially route anonymity or location privacy. In the rest of this section we mention few research done in the area of multicast anonymity.

Mutual Anonymous Multicast (MAM) presented in [Xiao 2006] designs a unicast mutual anonymous protocol and construction of an anonymous multicast tree-based communication. MAM is designed for wired networks. Here is the main idea: first the efficient multicast tree is formed by non-anonymous nodes

and then anonymous members connect to the available non-anonymous nodes. If there is not enough available non-anonymous nodes (they are saturated) then the anonymous nodes can connect to other anonymous nodes in the tree or a number of middle outsider nodes (who do not hide their identities) are invited to join the tree to be used as connection points if necessary. The anonymous members use a unicast initiator anonymous protocol to join non-anonymous members and use a unicast mutual anonymous protocol to join another anonymous member.

The unicast initiator anonymous protocol uses an onion for packet forwarding. The mutual unicast anonymous route between two anonymous members is formed by establishing two initiator anonymous unicast connection from each anonymous member to one of the middle outsiders and then using the middle outsiders as intermediate nodes to connect the two anonymous nodes.

One of few proposals for MANET anonymous multicast routing is EEAMA [Kao 2007]. EEAMA has been designed mainly with regard to energy efficiency to address the problem of limited capacity of batteries in mobile devices. In fact, EEAMA is put on top of any existing anonymous unicast routing protocol and therefore inherits its security features from the underlying protocol.

EEAMA's main contribution is to establish a multicast tree to enable more efficient data forwarding compared to unicast routing. Every receiver will establish a unicast route to the group source. On top of these unicast routes EEAMA builds the multicast tree considering special tree construction restrictions to keep up the security level achieved by the underlying unicast protocol. For example no node on the tree is allowed to be a branch node. This means even when a group receiver can join the group by finding a path to another node on the tree, it is required to discover a totally separate path to the source instead of using a tree member as a branch (except two neighbor group receiver). This property causes the receivers to discover and maintain redundant and sometimes longer routes to the source. However, due to these restrictions the constructed multicast tree might not be ideal which means packets could be forwarded in a more efficient way.

The other disadvantage of this protocol is that the group receivers' location information are attached to the join request and sent to the source node, so the source node is aware of the receivers' location and the receiver's location privacy is not fully provided. If the source is disclosed to the adversary or if some adversary can decrypt the join message it can find out the location of the group receivers. It furthermore means that the source node should have a secret key shared with each group member in advance.

Another approach for multicast anonymous routing protocol proposed for MANETs called AMUR was introduced by Lichun Bao in [Bao 2007]. AMUR uses Bloom Filters and Diffie-Hellman key exchange to provide anonymity. AMUR is actually an extension to the unicast anonymous protocol ODAR [Sy 2006] to multicast. A Bloom Filter is an enhanced hash table that uses

several hash functions. Such cryptographic tool is used as a data structure to test if an element is a member of a set or not (with a false positive rate and a false negative rate). In this protocol a Bloom Filter forms one part of the data packets, in which all of the source multicast tree links are stored. Also another Bloom Filter is included in the data packet structure containing the path from the sender to the intended receiver. The protocol aims to achieve different types of anonymity: Identity anonymity in AMUR means only neighboring nodes, i.e. nodes which do have a direct connection, know each others identity. In addition to that AMUR claims to feature location and routing anonymity, meaning that a node's location cannot be easily identified and a forwarding node cannot identify any other nodes en route including the source and recipient nodes. AMUR encrypts all links using secret keys established via a separate key exchange protocol based on Diffie Hellmann Key Exchange [Rescorla 1999] before storing them in the bloom filter. Topology information (information about current neighboring nodes) is exchanged periodically. The authors focused on the Bloom Filter issues, i.e. false positive and false negative rates and did not perform any performance analysis. They did not evaluate the delivery ratio or data packet latency under different node mobilities or traffic loads. In neighbor protocol of AMUR each node knows one-hop neighbor information and identity. In this protocol denial of service attacks are possible because the malicious nodes on the source path are not prevented from injecting invalid packets.

4.3 AnoMul: A New Approach toward Anonymous Multicast Routing in MANETs

AnoMul is a mesh-based multicast routing protocol. The mesh is initiated by the first node deciding to receive the group data. Our goal is to extend the identity anonymity provided for unicast communication in ANODR protocol, [Kong 2003], to multicast scenarios as well as providing location privacy for the leader and senders of the group assuming the presence of an eavesdropper while the performance is preserved high. In the following sections we will describe the protocol design in details and then we will evaluate both privacy features and performance properties of AnoMul.

4.4 Protocol design

4.4.1 Network Model

The network is composed of a number of mobile ad hoc nodes belonging to the same group. A public/private group key is pre-loaded to every nodes in advance possibly by an off-line third party. The presence of the third party is not required after the network deployment. The radio links between nodes is assumed to be

symmetric in the sense that if node X is able to hear node Y then Y would also be able to hear X. The nodes are able to perform both unicast and broadcast communication in their radio range. They also need to be able to do symmetric and asymmetric encryption and decryption.

4.4.2 Attacker Model

In this work we assume the Kerckhoff Principle about the adversary, i.e. the adversary knows every methods being used in the network. We consider a passive external adversary who can be composed of a number of collaborative eavesdropping nodes. A passive adversary is interested in accessing the information about the identities of the group members and their locations and/or tries to link between these two properties of a node by tracing the packet flows.

Another attacker considered in this work is the adversary who has the ability of compromising group members. A compromised node is usually considered as an internal adversary. The adversary cannot capture a non-limited number of nodes. The computational abilities of the adversary are supposed to be limited, i.e. it cannot decrypt messages in a reasonable time.

To be more specific, the protocol aims to prevent the attacker from the following attacks: identifying the nodes' IDs, inferring the location of the group leader by identifying and tracing the special packet flow initiated by it and finally finding the location of the group senders by tracing the data packet flows hop-by-hop back to their origin.

In the following sections the design of the protocol including how the multicast mesh is formed and maintained as well as the designed privacy mechanisms will be described.

4.4.3 Group communication components

4.4.3.1 Mesh Construction

We use the identification-free route discovery approach, introduced by J. Kong and X. Hong in [Kong 2003], to form the mesh. If a group member decides to receive the group data packets, it should join the mesh by broadcasting a JREQ (Join REQuest) message and waiting for JREP (Join REPly) messages from current members of the mesh.

We would refer to a group receiver who is already attached to the mesh as a *mesh member group receiver* and to a group receiver who is not attached to the mesh as a *non mesh member group receiver*. JREQ and JREP packets in AnoMul are basically of the same format of route discovery messages in ANODR protocol but adapted to multicast routing. The JREQ message format is as follows.

$$< JREQ, TTL, seq\#, nonce, onion, PK - 1time >$$

The onion and the one time public key, $PK - 1time$, are generated as in original ANODR protocol as described in the second chapter. We add the Time To Live field, TTL, to the JREQ message which is set by the source of the packet and will be decremented at every intermediate node until it reaches zero. *nonce* is a one time random word generated by the packet source used to authenticate the node which responds to the JREQ message. The responding *mesh member group receiver* needs to prove to the JREQ initiator that he is a valid group member. For this purpose we suggest a mechanism using group signatures.

Group signatures: In a group signature scheme the group members can sign a message anonymously on behalf of the group. Every group member has its own private group key while the group public key is the same for the whole group. The receiver of a signed message can use the group public key to verify if the signature is valid, but it cannot discover which group member has signed the message. A valid group signature means that the signer is a group member. The signatures do not reveal the nodes' identifications. An important component of a group signature scheme is the group manager who has the responsibility of handling the memberships. The group manager runs some interactive protocol with any new group member which results in producing its secret membership key. Also in case of a dispute the group manager can discover which group member has signed the message using its group manager secret key. In older proposed group signature protocols the size of the keys was linearly dependent on the group size, but there are some proposed schemes such as [Camenisch 1998], [Camenisch 1997] in which the signature size is independent from the size of group and are efficient enough to be applicable for ad hoc networks.

Group signatures perfectly match to our need. There should be some group key establishment before the multicast session starts, i.e we assume that the group key is pre-loaded to the nodes before the deployment of the network. It can be done by a group manager whose responsibility is just the group key pre-establishment. We suppose that the private keys and the public key do not change during the multicast session.

When a *mesh member group receiver* receives a JREQ message first it will check if the seqence number has not been seen before. If so it will reply with a JREP message with the probability of P_{rep}. This is done to avoid too many JREP packets triggered on arrival of the JREQ packet at every *mesh member group receiver*s. If the node is going to reply to the received JREQ packet it would sign the received nonce using its group private key and would send it back to the JREQ initiator. The JREQ initiator will try to open it using the group public key and if the decrypted nonce matches to the original *nonce* sent before, it means the JREP initiator is a valid group member (a valid signer) because no one else has access to a group private key. The JREP message format is as follows.

$$< JREP, \{K_{seed}\}_{PK-1time}, f_{K_{seed}}(GK_i(Nonce), onion) >$$

GK_i is a group private key held by the JREP message source and f is a one way function. The JREQ sender will open the signed *nonce* and compare it to the original one to make sure that the JREP comes from a valid group member. Using this method the group members the node is using as a port to attach to the mesh (the ones who send back a JREP) are authenticated.

Upon receiving the first JREP message the node considers itself as being connected to the mesh. Such a node would be called a *knot*. Actually a *knot* is defined as a group member who is already joint to the mesh. The node sets its internal flag to '*knot*' after successful joining. If the joining node does not receive any JREP message during some specified waiting time it will broadcast a new $JREQ$ with an increased TTL as it is described in 4.4.3.3. If the joining node receives more than one valid JREP messages it will join the mesh through the first two ones. In section 4.4.3.9 we will present our analysis about the optimum number of connections discovered by any joining node to the mesh regarding both mesh connectivity and overhead.

This property in combination with the fact that in AnoMul every *knot* responds to a received JREQ packet with the probability of P_{rep} cause that the two discovered routes to the mesh are neither necessarily the shortest ones nor very close to each other. It will give the protocol a richer connectivity and a higher robustness, because if the two discovered routes are not very close to each other it will not be very likely both of them are lost due to topology changes or any other local reason. Note that as two JREQ packets with the same sequence numbers would not be processed by a ginev node, thus the two routs found by one node joining the mesh will be disjoint.

4.4.3.2 Leader Election Mechanism

If the joining node receives at least one JREP messages it realizes that a multicast mesh is already constructed for that group and it joins it through the discovered routes. If it does not receive any JREP for its JREQ message it will try again with an increased TTL value as explained later. The node will do this up to three times in the case of not receiving any reply from the mesh. If after the third JREQ broadcast no reply is heard from the mesh the node will consider itself as the first mesh member and therefore the mesh *leader*. The next node that needs to join the mesh will find its route to the leader. As already mentioned, once being joint to the mesh a group receiver is called a *knot*. Then the next joining receivers would join the group by finding their routes to the existing *knot*s on the mesh. The general structure of the network is shown in Figure 4.3 in which the mesh is composed of the *leader*, the *knot*s and the intermediate nodes on the routes.

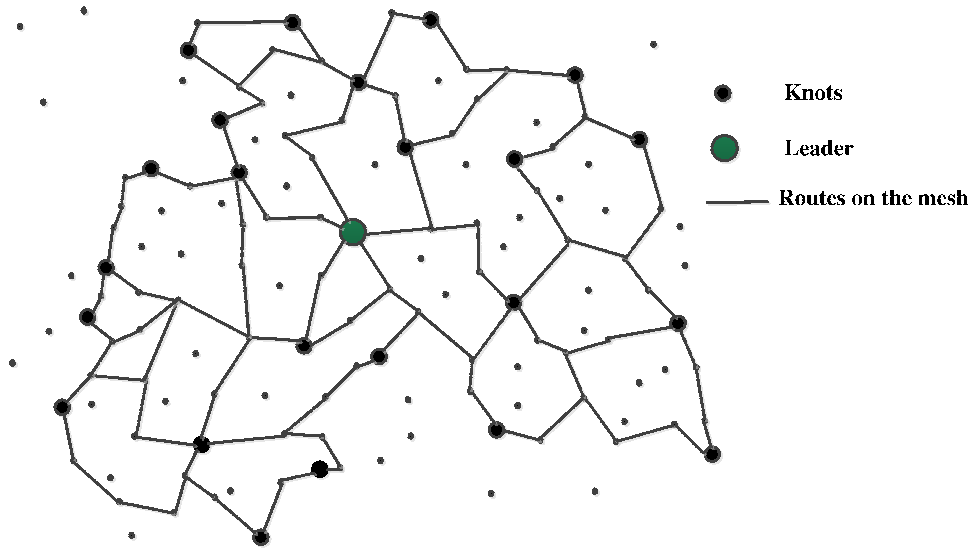

Figure 4.3: The Mesh Structure

4.4.3.3 Setting the Time To Live field in JREQs

We need to have a proper estimation for the TTL in JREQ packets to prevent the packet from being flooded through the whole network. It is desirable the packet will reach at least two knots of the mesh but not many more. In Figure 4.4 S represents the whole network area and S' is the area in which the JREQ packet propagates (till the TTL reaches zero). R is the whole network area in terms of the number of present nodes. If $TTL = \frac{R}{n}$ then we have $S' = \frac{1}{n^2}S$. Let K be the number of *knot*s in the network at the present time. Then supposing a uniform distribution of the nodes we will have:

$$N_{Rep} = \frac{P_{rep}.K}{n^2} \tag{4.1}$$

N_{Rep} is the desired number of reply packets initiated in response to a JREQ message, which, as will be described in 4.4.3.9, would be $N_{Rep} = 2$. P_{rep} is the probability that a *knot* replies to a received JREQ message as explained in section 4.4.3.1. K could be either known as a statistical property of the network or estimated by the group member according to the current application or the network status. n would be calculated from the above equation as

$$n = \sqrt{\frac{P_{Rep}K}{N_{Rep}}} = \sqrt{\frac{P_{Rep}K}{2}} \tag{4.2}$$

and the required TTL will be found from

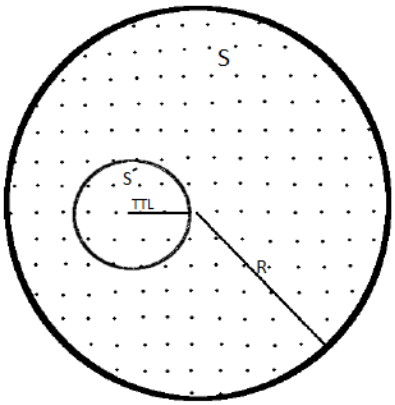

Figure 4.4: TTL estimation in the JREQ packets

$$TTL = \frac{R}{n} = \frac{R}{\sqrt{\frac{P_{Rep}K}{2}}} \tag{4.3}$$

If no JREP is received or the number of received JREPs is less than the desired amount, the node will increase the TTL in its new JREQ packet. Such *knot* will suppose that the number of *knot*s in the mesh is less than what was assumed in the last iteration and therefore it will try again to find routes to the mesh with a smaller K value. $K_2 = K_1.\frac{1}{k}$ will result in $n_2^2 = \frac{n_1^2}{k}$. So the updated TTL would be $TTL_2 = \sqrt{k}.TTL_1$. The node has to increase k from 1 to maximum n and stop when the leader is reached. How fast k is increased could be decided based on the application, i.e. if a lower risk of latency is more important than a lower risk of overhead k has to be increased faster and vice versa.

4.4.3.4 Message Type Unification Mechanism for non-route-discovery Messages

We design a mechanism to unify the packet types for non-route-discovery messages (the messages not intended for route discovery) named *Ptype* which stands for Protected TYPE messages. The purpose of this mechanism is to prevent the adversary from distinguishing between some specified messages in order to hide different node activities.

For example as one of the message types on which this mechanism would be applied is leave messages, the eavesdropping adversary will not be able to recognize leave messages from their appearance and therefore he cannot realize

that a node is leaving the mesh. It would also facilitate providing some important privacy mechanisms such as group leader location privacy by hiding the leader messages as we will see later.

The idea (similar to the modification applied to route request and route reply packets in RDIS in the previous chapter) is to make the appearance of the *Ptype* packets to look the same. For instance there should not be a fixed field only in one of them (which does not change over different hops the packet traverses) while the same field may change in the other packet types since such a field will be a clue for the adversary to distinguish that kind of packet among all *Ptype* packets. It is also important to assure different *Ptype* messages have the same size to prevent the adversary from distinguishing between them based on their different sizes.

Actually, the common property of *protected type* messages is that their content is encrypted with the shared secret of the link and if the node receiving the packet succeeds to decrypt it using the corresponding link pseudonym from its routing table it will need to proceed with the message no matter if the node does not know of which type it is. So although the type of such packets will not be carried in clearby it, no problem would arise due to the unknown type of the message at the nodes receiving them since they know the secret which should be used to discoer the type of packet.

4.4.3.5 Route Confirmation Mechanism

As described before, a node will join the mesh via the first two offered connections from the mesh. The recorded pseudonyms at the nodes on the rest of discovered routes will never be used and therefore need to be deleted from the routing tables. This goal is achieved by a confirmation mechanism. In this mechanism the JREQ initiator sends a *Route Confirmation message* over the first two routes indicating to preserve the recorded pseudonyms at all intermediate nodes's routing tables. The intermediate nodes which do not receive such a message (during some specified period after the route is formed) will remove the corresponding pseudonyms from their tables. The confirmation message is one of the Ptype messages as shown below.

$$< Ptype, N_i, f_{K_{seed}}\{\ i, confirmation, K_{seed}\} >$$

Like in unicast ANODR, K_{seed} is used as a seed to generate a route pseudonym sequence at each hop. N_i is the route pseudonym generated by applying a one-way function, f, i times on K_{seed}, where i is the increasing number of packets transmitted over the link.

$$N_i = f^i(K_{seed}) \tag{4.4}$$

The node receiving such a packet checks its routing table for this route pseudonym. If it is found, the node realizes this message is intended for it

and after decrypting it using the corresponding K_{seed} the node finds out about the message type and the revealed payload tells the node to save the recorded pseudonym. The nodes en route will modify the pseudonym field in the message to the next hop's pseudonym and forward the message on the route. If an intermediate node does not receive such a message during T_{conf} seconds it should delete the corresponding route pseudonyms from its routing table. T_{conf} is the maximum time that may elapse to receive the confirmation message which is chosen proportional to the maximum route length. This function prevents the routing tables from preserving entries which will never be used later in multicast data forwarding phase.

4.4.3.6 Data Forwarding

If a group sender decides to send some data packets to the group it will need to join the mesh by broadcasting a JREQ packet as described before. So the adversary cannot differentiate between a group sender and a group receiver by observing the signaling when they are joining the mesh. After the group sender has joined the mesh it can send data packets over the discovered routes to the group. When a *knot* receivs a data packet it would forward it over all routes it has to the neighboring *knot*s using the link pseudonyms already stablished. Actually based on the *United Pseudonym Mechanism* explained below each *knot* would just broadcast the data packets with the shared route pseudonyms and as the result every neighboring *knot* of it will receive the packets. In this way every *knot* in the network will receive every data packets transmitted by a group sender. The data packet format is as follows.

$$< Data, N_i, f_{K_{seed}}\{i, step\ counter, index, payload\} >$$

The data packets are numbered by *index* such that the nodes can use the *index* to reorder the data packets if they are not received in the right order. Every node forwards a data packet only if it has not forwarded a data packet with the same index before. The *step counter* is related to the source location privacy mechanism described in section 4.4.4.1.

4.4.3.7 Multiple Forwarding Issue: optimizing the packet forwarding at *knot*s

Let K_{seed} and K'_{seed} be the route pseudonyms saved at a *knot* for the first link on its discovered routes to the mesh. The *knot* might get connected to the mesh through more number of connections too, in the case that other *knot*s find their routes to the mesh through this node. Therefore it is very likely to happen that a given *knot* has several pseudonyms shared on its different connections to the

mesh. This means the *knot* would need to forward every data packet several times (each time with one of its shared route pseudonyms), which results in a high overhead of multiple forwarding at each *knot*. To avoid such a problem we introduce the *united route pseudonym* mechanism as follows.

United Pseudonym Mechanism: The idea is to replace all route pseudonyms stored at a *knot* with a common pseudonym to be used for data packet forwarding. Let's K_{seed} be the shared pseudonym of a given *knot* on its first discovered route to the mesh. When the *knot* receives the second join reply message from the mesh and sends a confirmation message over that route it should change the first hop's route pseudonym of that route, K'_{seed}, to K_{seed}. This is performed by sending a *modification message* on the first hop on that route of the following format.

$$< Ptype, N'_i, f_{K'_{seed}}\{i, modification, modify\ K'_{seed} \text{ to } K_{seed}\} >$$

Since it is enough that each route pseudonym is synchronized between the two nodes sharing it, this modification would cause no problem in packet forwarding over the routes. As a matter of fact, the original pseudonym shared on such a link would be used in the opposite direction (from the first node on the route toward the *knot*).

Later on, when another group member establishes a connection to this *knot* in order to join the mesh, the *knot* will do the modification mechanism on the new established route as well (will modify it to the current element of its united route pseudonym chain).

Therefore, a given *knot* would have a common route pseudonym shared with all of its connections and if the *knot* forwards a received data packet only once using its *united route pseudonym* all of its neighboring *knot*s will receive it.

4.4.3.8 Mesh Maintenance

The mesh *leader* maintains the mesh memberships by sending hello messages to the *knot*s. If a *knot* receives such a packet from the leader it realizes that it is still connected to the mesh. A leader hello message is shown below.

$$< Ptype, N_i, f_{K_{seed}}\{i, m_{leader}, t\} >$$

The time stamp shows the time at which the message has been issued. A *knot* will forward a received leader hello message only if it has not seen a leader hello message with the same time stamp before. If the time stamp has been seen already by the node, the message will be discarded as a duplicate. Every *knot* uses its *united route pseudonym* to forward leader messages.

The time difference between two consequent hello messages is chosen as a random variable uniformly distributed in the range of $[0, T_m]$. If a *knot* does not receive the hello messages it would realize that it has lost its connections to the mesh. Then it would discover new connections to the mesh to re-establish its connectivity to the mesh. If it has been receiving no leader messages for more than $3T_h$ seconds it should perform the JREQ procedure once again.

4.4.3.9 Mesh Connectivity Analysis

In this section, we analyse the mesh connectivity of AnoMul based on the designed mesh construction and maintenance described earlier. Briefly the mesh in AnoMul is formed as follows. The first node deciding to be a member of the mesh would be the leader. From the second node on, every new member will join by trying to discover the shortest possible two routes to the mesh. Every node will re-establish its connections to the mesh when it needs.

The number of routes established by each joining node would affect both mesh connectivity and performance. The more redundant the connections of each *knot*, the richer the connectivity of the mesh. On the other hand, each connection establishment means an increase in the routing overhead of the protocol. What we analyse in the following simulations would be the trade off between the mesh connectivity and the overhead caused by the number of joining and re-joining operations to the mesh by *knots*. We performed this simulations in Matlab.

Let N_{join} be the number of routes found to the mesh by each *knot* at the time of joining. We consider a network with 30 group members joining the mesh one after another. The nodes are initially distributed randomly in a $3000m \times 3000m$ network and are moving in the mesh with random way point mobility model. Every *knot* rejoins the mesh when it realizes it has been having no connectivity to the mesh for a specified number of leader hello periods.

We calculate the mesh connectivity, defined as the number of *knot*s which are connected to the mesh divided by the whole number of group members averaged over the simulation time, for different N_{join} values and for increasing node mobilities.

$N_{join} = 1$ is a special case, modelling a tree based multicast group instead of a mesh. In tree based multicast communication every member finds one route to the group. Although tree based protocols are bandwidth-efficient but they do not usually offer sufficient robustness.

The results are shown in Figures 4.5 and 4.6 as the mesh connectivity and the routing overhead in terms of number of join/rejoin operations for $N_{join} = 1$ to $N_{join} = 5$. As Figure 4.5 shows, the mesh connectivity of the tree based topology, as expected, is far below the connectivity of mesh based scenarios. Although, as Figure 4.6 shows, the topology construction overhead of tree based approach is the lowest one but compared to its very poor connectivity when the nodes are moving it does not worth to be chosen for group communication.

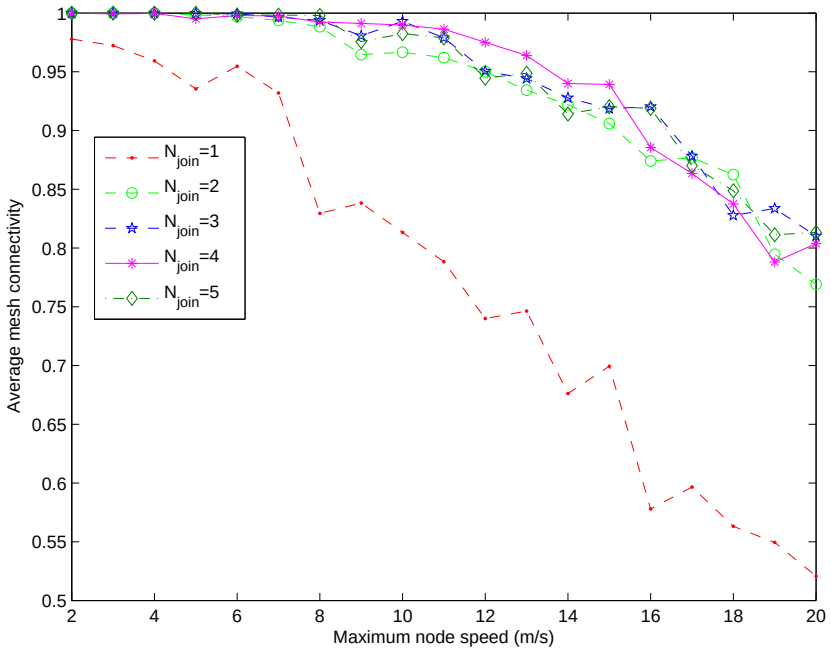

Figure 4.5: Mesh connectivity for different N_{join} values

Comparing the connectivity and overhead of mesh scenarios with $N_{join} = 1$ to $N_{join} = 5$, we conclude that the optimum case would be $N_{join} = 2$. This is because regarding connectivity all of the mesh approaches give almost the same result while the routing overhead grows dramatically with N_{join}. Therefore we design AnoMul based on finding two connections toward the mesh by each joining group member, to optimize the trade off between the mesh connectivity and the routing overhead.

4.4.3.10 Mesh Member Leave

If a node decides to leave the mesh it will send a *leave announcement message* to all of its neighboring *knots* using its *united route pseudonym*. A *leave announcement message* is of the following format.

$$< Ptype, N_i, f_{K_{seed}}\{i, leave, knot\ leave\} >$$

A leave message will be forwarded over the routes towards the neighboring *knots*. On receiving such a message the intermediate nodes delete the correspondent route pseudonyms from their routing tables. When a neighboring *knot* receives this message it will drop it as it is the last node supposed to read such a

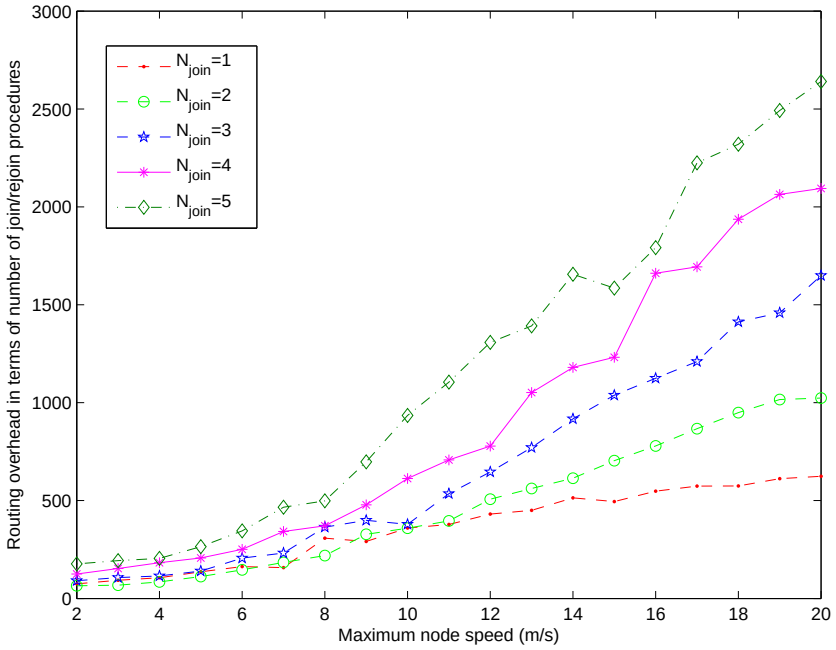

Figure 4.6: Join/rejoin overhead for different N_{join} values

message. Then it waits for T_h seconds to receives the next leader hello message. If it does not hear anything from the leader, it will realize it has no connection to the mesh any longer and it will rejoin the mesh.

If the leader decides to leave the mesh, before sending the leave message it should choose one of its neighboring *knot*s as the new leader. For this purpose it broadcasts a JREQ message. After receiving the first JREP message it sends a confirmation message and then a *leader leave message* of the following format over the first discovered route.

$$< Ptype, N_i, f_{K_{seed}}\{i, leave, \ leader \ leave\} >$$

If the *knot* which receives the above message agrees to be the mesh leader it will send back a message of the following format to confirm it is going to be the new leader.

$$< Ptype, N_i, f_{K_{seed}}\{i, leave, \ new \ leader\} >$$

Otherwise the leader needs to choose another *knot* as the new leader repeating the same process again. The above messages are sent using the original k_{seed}s of

the links to direct it only to the single desired *knot*. After choosing a new leader the ex-leader sends a normal *leave message* over its connections as described above and stops generating leader hello messages.

4.4.4 Location privacy Mechanisms

An adversary who aims to identify the network members may try to disclose a node's identity or its location. In this approach due to using the identification free routing proposed in [Kong 2003] the nodes' identities cannot be discovered by the adversary, so we concentrate on the nodes' location privacy.

4.4.4.1 Group Sender location Privacy

To protect group sender's location privacy, our solution is to step away the source behavior from the real sender. In this mechanism for each data packet the group sender selects one of its neighboring *knot*s randomly and unicasts the packet to it. The data packet contains a step counter which is a number, n, chosen randomly in the range of [2,n_s]. After receiving that packet, the neighboring *knot* will decrement the step counter by one and forward the packet randomly to one of its neighboring *knot*s after a random short delay. This process will continue till the step counter reaches zero at one *knot* who would multicast the packet to the mesh. Such a *knot* is called a $fake\ source$. We name this behaviour *stepped source* mechanism. n_s is selected according to the required privacy level, the larger the n_s the higher the privacy level as showed in section 4.5.2.

This way, every data packet will traverse a random way to a random *knot* before being multicast. The considered attacker is the one who tries to reach the source node by tracing the data packets over consecutive hops (over next hop it will need to trace the source of next comming packet in the flow). The described mechanism prevents such an adversary from performing this attack because each data packet will be sent over a different path before being broadcast. As Figure 4.7 shows in stepped source mechanism each data packet would be transmitted to a fake source through a unicast path.

4.4.4.2 Group Leader Location Privacy

Location privacy of the group leader is one of the important privacy aspects in multicast protocols. If the adversary finds the location of the leader, it could perform denial of service attacks or node capture attack against it which could cause the mesh to fail. We provide the mesh leader location privacy exploiting the combination of what we explained in 4.4.3.8 about the leader hello messages' generation intervals and the message type unification mechanism. The hello messages cannot be identified by the attacker from their appearance since hello messages are one of the packet types to which the message unification mechanism is applied. On the other hand, as mentioned the leader initiates leader

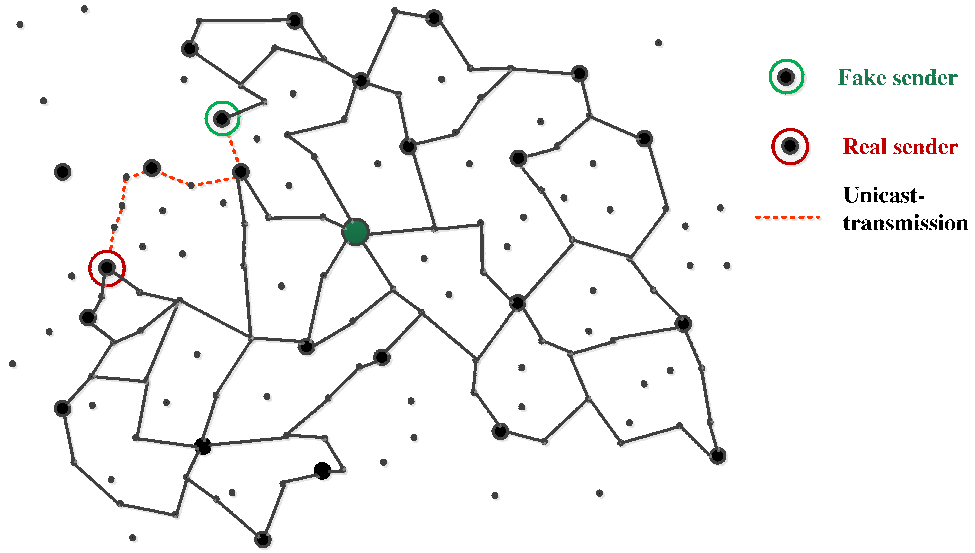

Figure 4.7: Stepped source mechanism for one data packet (n=3)

hello messages in randomly chosen intervals in the range of $[0, T_m]$ instead of periodically generating them. This prevents the eavesdropping attackers from correlating timing of leader message transmissions.

If the hello messages are generated periodically then the strong enough eavesdropper could trace any periodic packet flow patterns to find their initiator. The initiator of such packets would be considered to be the mesh leader. Therefore, in our approach the adversary is not able to recognize the location of the leader by tracing back the leader hello messages.

4.5 Privacy Analysis

Due to the identity-free approach in AnoMul no node ID would be revealed to the outsiders. In addition the *knot*s are anonymous to each other in routing because given two encrypted commitment values (signed nonces) it is impossible to decide which group members have signed them or whether they are signed by the same or different group members due to the common property of group signatures.

Another privacy property of AnoMul is hiding the network activities from the eavesdroppers because of using *Ptype* mechanism. For example, the adversary would not realize when a node leaves the mesh or a discovered route is confirmed.

The group senders and the group leader's location privacy will be analysed in the following sections.

4.5.1 Leader Location Privacy

Due to hiding the leader hello messages among other *Ptype* messages the adversary would not be able to distinguish every single hello message to be able to track their source to find out where the mesh leader is located. If the adversary is strong enough in overhearing the messages in the network field he could be able to analyse the traffic to find the ongoing traffic patterns. If the hello packets are periodic, as in most multicast protocols, then a strong enough adversary having enough number of colluding eavesdropping nodes all over the network could be able to determine the hello messages' interval, let's say T_h. Being aware of T_h an overhearing node could start from a point located in a low activity area of the network and follow the periodic *Ptype* packets arriving every T_h second hop by hop toward the group leader.

As mentioned in section 4.4.4.2 in AnoMul T_h is chosen as a uniform random variable in $[0, T_m]$. Therefore, it is more difficult for the adversary to find the group leader because even if he knows that $T_h \in [0, T_m]$ he has no idea which *Ptype* packets he should follow since obviously not all *Ptype* messages are hello messages. Therefore, at some steps toward the leader the adversary might follow wrong *Ptype* packets which are in fact of other kinds of *Ptype* messages. This increases the uncertainty of the adversary about the leader's venue when he is performing the above attack.

In this section we name hello messages as 'red *Ptype*' and call the other kinds of *Ptype* packets as 'black *Ptype*' packets. Obviously the red *Ptype* packets lead the eavesdropper towards the group leader and the black ones would mislead him. On average a red *Ptype* message arrives at a given node in the network every $\frac{T_m}{2}$ second. On the other hand the average rate of black *Ptype*s depends on the amount of group activities and nodes' mobility. Let us assume a given node receives a black *Ptype* packet on average every T_b second. Therefore the first overheard *Ptype* packet at a node could be a red one with the probability of $P_r = \frac{2T_b}{T_m+2T_b}$ and could be a black one with the probability of $P_b = \frac{T_m}{T_m+2T_b}$. The more uncertain the adversary is about the *Ptype*s to be red *Ptype* packets the more uncertain he would be to believe that he is getting close to the group leader's venue when he is following the *Ptype* messages.

As mentioned in section 4.4.2 based on our assumption about Kerckhoff's Principle, the adversary is aware of the *Ptype* mechanism designed against him in AnoMul. So the only thing that an eavesdropping attacker can do would be to follow the first arriving *Ptype* message at the point it is eavesdropping to its origing and do the same at the new point (although he is uncertain if it is a red or a black *Ptype* packet). The adversary can establish several such eavesdropping nodes in different locations in the network field. Tracking back the *Ptype* messages, most of such eavesdropping nodes would reach close to each other in some area in the network in long term where the adversary will suppose the group leader is located somewhere nearby.

We introduce a mathematical model to calculate the probability that the adversary will reach the group leader under above attack in a given period of time. If there were no black *Ptype* packets, i.e. $P_b = 0$ and $P_r = 1$, then any eavesdropping node would reach the leader through the shortest path from its initial location to the leader.

Let us consider a random walk in one dimension first. Suppose in every step the walker moves one step towards the goal point with the probability of P_1, moves one step away from it with the probability of P_2 or does not move at all with the probability of $1 - P_1 - P_2$. Therefore, each step is a random variable, x_i, with the mean of $\mu_i = P_1 - P_2$ and the variance of ${\sigma_i}^2 = E({x_i}^2) - {\mu_i}^2 = P_1 + P_2 - \mu_i^2$. According to the central limit theorem after many steps, let's say l steps, the final position of the walker, $X = \frac{x_1+x_2+...+x_l}{\sqrt{l}}$, can be estimated with a normal random variable.

$$X \longrightarrow \mathcal{N}(\sqrt{l}\mu, l\sigma^2) \tag{4.5}$$

Therefore the probability that the walker has not reached the goal point after l steps would be:

$$P(X \leqslant \frac{D}{\sqrt{l}}) = \frac{1}{\sqrt{2\pi l\sigma^2}} \int_0^{\frac{D}{\sqrt{l}}} e^{\frac{-(X-\sqrt{l}\mu)^2}{2l\sigma^2}} dX \tag{4.6}$$

where D is the initial distance from the walker to the goal point. Actually if P_b is high enough the attacker could reach the leader with a big delay if ever. The probability that the first overheard *Ptype* packet arriving at a given node is a black one coming from any of its four neighboring nodes located up, down, left or right to it is $\frac{P_b}{4}$. On the other hand, since hello messages are propagated away from the leader we assume that a node receives the (first) hello message in each hello round from the neighbor which is closer than it to the leader. We call this neighboring node as the 'prior' neighbor of the node. Therefore, the probability that the first overheard *Ptype* message arriving at a given node is a red one sent by the prior neighbor of it is P_r and the probability that it is a red one sent by any of the other three neighbors is zero (See Figure 4.8). So, at each point the first *Ptype* packet comes from the prior neighbor with the probability of $\frac{P_b}{4} + P_r$ and comes from any of the other three neighboring nodes with the probability of $\frac{P_b}{4}$.

If the possible path from the eavesdropper toward the leader was single (the shortest path) we could say the attacker steps towards the leader with the probability of $P_1 = \frac{P_b}{4} + P_r$, steps away from it with the probability of $P_2 = \frac{P_b}{4}$ or does not move with the probability of $2 \times \frac{P_b}{4}$. This is the attack model in a one dimensional network which is a one dimensional random walk (a network in which the only path from the adversary to the leader is a single path).

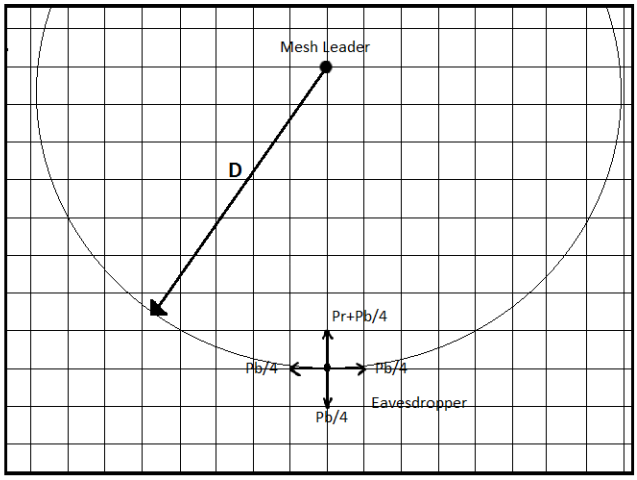

Figure 4.8: The attacker traces the *Ptype* packets to find the leader

In the real two dimension network the attacker could easily get deviated from the shortest path from his current position to the leader by the next black *Ptype*s. But every time that it happens, the new position of the attacker has exactly the same properties of the previous one, i.e. it has the same movement probabilities. Therefore to estimate the distribution of the final position of the attacker we can use the same random distribution of a one dimension network. So each step in such a random walk is a random variable with the mean of $\mu_i = P_1 - P_2 = P_r$ and the variance of ${\sigma_i}^2 = P_1 + P_2 - \mu_i^2 = \frac{P_b}{2} + P_r - P_r^2$. Actually, the two dimensional model made from the one dimensional model is valid because each time the adversary deviates from its straight path toward the leader (by moving to right or left in Figure 4.8), it's distance to the leader is not changing (this movement is estimated as moving on the circle in Figure 4.8 which can happen in two different directions with the same probabilities). Therefore each eavesdropping node will reach the leader after l steps with the probability of:

$$P(X \geqslant \frac{D}{\sqrt{l}}) = 1 - \frac{1}{\sqrt{2\pi l(\frac{P_b}{2} + P_r - P_r^2)}} \int_0^{\frac{D}{\sqrt{l}}} e^{\frac{-(X-\sqrt{l}P_r)^2}{2l(\frac{P_b}{2}+P_r-P_r^2)}} dX \tag{4.7}$$

Figure 4.9 presents the probability that the adversary can succeed to find the group leader's venue as a function of number of its tracing steps while following hello messages. For $D = 12$ as this graph reports, after D steps the adversary is very unlikely to find the leader and even after $3D$ steps the probability is

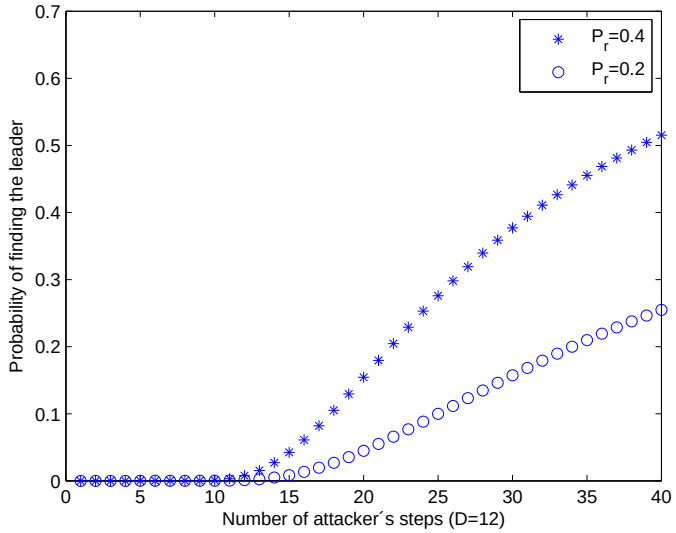

Figure 4.9: The probability that the attacker finds the leader

still not high. Of course the higher the P_r the higher the probability of leader's disclosure.

4.5.2 Sender Location Privacy

In stepped source idea the fake source is chosen as the destination of a random walk from the real source. As described in section 4.4.4.1 the step number, n, is a random number in the range of $[2, n_s]$. The sender location privacy level will depend on the range in which n is chosen. The smaller the n_s the closer the fake sources to the real source. If most of the fake sources were nearby the real one, a strong enough adversary could be able to determine the area where most of the data packets are started from, which most likely includes the real source's venue.

If $n \in [2, n_s]$, the average value of n will be $\bar{n} = \frac{n_s+2}{2}$. The walk steps can happen in any direction randomly. So we consider a grid with the real source in the center of it where the random walk starts with an average step number of $\bar{n}$. At each step the packet could move up, down, left or right with the same probabilities of $\frac{1}{4}$. We consider the movement as the summation of two random walks, one in x dimension and the other one in y dimension. Since the node does not move horizontally if it moves up or down and it does not move vertically if it moves left or right, so we can model the whole movement as follows. In its horizontal (vertical) random walk at each step the packet does not move at all with the probability of $\frac{1}{2}$ and moves to right or left (up or down) each with the probability of $\frac{1}{4}$. So the movement in each dimension is a random variable with

the mean of zero and the variance of $\sigma_x^2 = \sigma_y^2 = \frac{1}{2}$. If the number of steps is high the final position of the packet in each dimension can be estimated with a Gaussian random variable according to central limit theorem:

$$\frac{x_1 + x_2 + ... + x_{\bar{n}}}{\sqrt{\bar{n}}} \sim \frac{1}{\sqrt{2\pi {\sigma_x}^2}} e^{-\frac{x^2}{2{\sigma_x}^2}} \tag{4.8}$$

$$\frac{y_1 + y_2 + ... + y_{\bar{n}}}{\sqrt{\bar{n}}} \sim \frac{1}{\sqrt{2\pi {\sigma_y}^2}} e^{-\frac{y^2}{2{\sigma_y}^2}} \tag{4.9}$$

Therefore we estimate the final position of the random walk as:

$$\frac{|\vec{r_1} + \vec{r_2} + ... + \vec{rn}|}{\sqrt{\bar{n}}} \sim \frac{1}{\sqrt{2\pi {\sigma_x}^2}} e^{-\frac{x^2}{2{\sigma_x}^2}} * \frac{1}{\sqrt{2\pi {\sigma_y}^2}} e^{-\frac{y^2}{2{\sigma_y}^2}} = \frac{1}{\pi} e^{-(x^2+y^2)} \tag{4.10}$$

where $\vec{r_i}$ means the ith random walk step. If R is the final position of the random walk we have:

$$P(\frac{|\vec{R}|}{\sqrt{\bar{n}}} \leq \frac{d}{\sqrt{\bar{n}}}) = \frac{1}{\pi} \iint\limits_{0 \leq \sqrt{x^2+y^2} \leq \frac{d}{\sqrt{\bar{n}}}} e^{-(x^2+y^2)} dx dy \tag{4.11}$$

By replacing $x' = \sqrt{\bar{n}}x$ and $y' = \sqrt{\bar{n}}y$

$$P(\frac{|\vec{R}|}{\sqrt{\bar{n}}} \leq \frac{d}{\sqrt{\bar{n}}}) = \frac{1}{\bar{n}\pi} \iint\limits_{0 \leq \sqrt{x'^2+y'^2} \leq d} e^{-\frac{1}{\bar{n}}(x'^2+y'^2)} dx' dy' \tag{4.12}$$

$$P(|\vec{R}| \leq d) = 1 - e^{-\frac{d^2}{\bar{n}}} \tag{4.13}$$

The fact that the distribution of the fake sources's location is Gaussian with the mean of zero means many of them will be located more or less close to the real source. This could give a clue about the real source's location to the strong eavesdropper which is monitoring the traffic all over the network. To calculate the adversary's uncertainty about the real location of the source node we assume there is one source node in the network. The adversary overhears the message flows in the network to detect how many data packets are originated from different locations in the network. According to the above equation the area with radius of d hops around the real source will include $p = 1 - e^{-\frac{d^2}{\bar{n}}}$ percent of fake sources on average during the time. When the adversary discovers such an area then from his point of view any node inside that area could be suspected

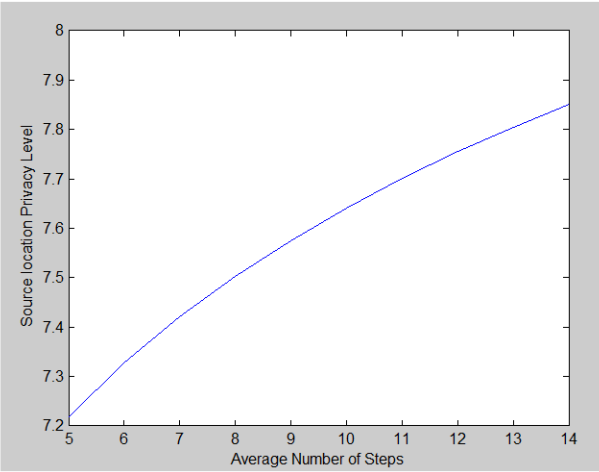

Figure 4.10: Adversary's uncertainty about the sender venue as a function of $\bar{n}$

to be the source node with a probability of $P_{in} = \frac{p}{N_{in}}$ and any node out of that area could be the real source with probability of $P_{out} = \frac{1-p}{N-N_{in}}$ where N is the whole number of nodes in the network and $N_{in} = D\pi d^2$ is the number of nodes in the detected area in where D is the node density of the network. According to the definition of Shannon entropy the adversary's uncertainty about the location of the group sender is as follows:

$$H = -\sum_{1 \leq i \leq N_{in}} P_{in} log_2(P_{in}) - \sum_{1 \leq i \leq N_{out}} P_{out} log_2(P_{out}) \tag{4.14}$$

For a network field of $2km \times 2km$ with $D = 100\ nodes/km^2$ Figure 4.10 shows the adversary's uncertainty about the real location of the group source on average as a function of $\bar{n}$. To have a more clear sense of the level of source location privacy we compare the uncertainty level with a uniform distributed random variable. If M events could happen with the same probability the uncertainty about the event is $log_2(M)$. So $H = 7.5$ for source node location privacy is equal to the case that $2^{7.5}$ nodes are equally likely to be the group sender.

Now let us assume the adversary has the ability to capture a limited number of nodes. His uncertainty about the source venue would decrease since he is certain about the captured nodes. Figure 4.11 presents how the fraction of captured nodes decreases the source location privacy.

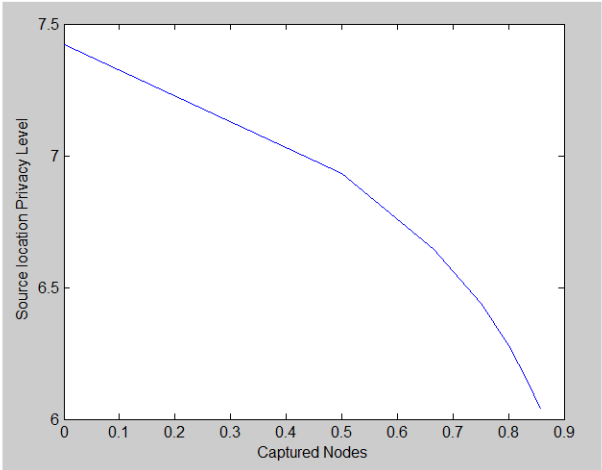

Figure 4.11: Adversary's uncertainty about the sender's venue when some nodes are betrayed

4.6 Protocol evaluation

We implemented AnoMul in Qualnet, a simulator for wireless and wired networks by Scalable Network [Qua], to evaluate its performance. We evaluate the performance of the protocol in terms of packet delivery ratio, end-to-end delay, jitter and routing overhead in different scenarios. The delivery ratio is computed as the sum of all received data packets divided by the sum of all sent data packets (whole number of transmitted data packets is considered as the number of sent data packets by the group senders multiplied by the corresponding number of group receivers). The end-to-end delay is the average time difference from sending to receiving the packet at the group receivers. Jitter is calculated corresponding to the formula given in RFC1889 for RTP [RFC]:

If S_i is the RTP timestamp on packet i, and R_i is the time of arrival for packet i, then for the two packets i and j the parameter D is defined as:

$$D(i,j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \tag{4.15}$$

The inter-arrival jitter of the data stream is calculated continuously as follows, using D for each packet i and the previous packet $i-1$ in order of arrival (not necessarily in sequence):

$$J = J + (|D(i-1,i)| - J)/16 \tag{4.16}$$

In real time applications such as video streaming jitter could be more important than delay itself and in some applications it is controlled using playout

buffers to prevent the quality of service to degrade [Wang 2008].

Routing overhead is given by the sum of bytes of routing packets divided by the sum of bytes of data packets successfully delivered to the group receivers.

We will study the influence of node mobility, group size and traffic load (number of group senders) on above performance metrics. Also, in each case we set-up a similar scenario in which ANODR is used to deliver the data packets to a group of receivers. Comparing AnoMul and ANODR simulation results can show how successful AnoMul is performing as a multicast routing protocol to manage the group-based communication efficiently compared to the underlying unicast routing protocol.

4.6.1 Simulation Model

The simulation model used to evaluate the protocol performance is as follows, unless otherwise mentioned. 60 nodes with uniform distribution are placed on a $1000m \times 1000m$ area. All experiments use a Constant Bit Rate (CBR) generator application to generate packets. Each data packet has a size of 512 bytes. Data packets are generated at the group senders with a rate of 4 packets per second. The overall simulation time is 3600 seconds. Each node is equipped with a single antenna using 802.11b with default parameters with a transmission range of about 350 meters. Node mobility is simulated using the random waypoint model with pause times of 30 seconds.

4.6.2 Simulation Results

First, we set up a multicast session consisting of 60 nodes and 30 group receivers which have been selected randomly and join the mesh one after another. We run the simulation to check the data packet delivery ratio, the end-to-end delay, jitter and the routing overhead caused by the protocol. Figure 4.12 shows the delivery ratio of AnoMul with 30 receivers for node speeds of zero to 10 m/s as well as the same parameter when ANODR is used to deliver data packets to the group of receivers. As this figure is reporting, AnoMul is performing well in delivering data packets to the mesh receivers and at the high speed of 10 m/s it is still delivering more than 95% of data packets successfully. The gap between the delivery fraction of AnoMul and ANODR grows as the mobility increases.

In AnoMul even at high mobilities, the good connectivity of the mesh prevents having a lot of packet loss, i.e. when due to the high mobility a number of routes fail the data packets will still have a good chance to be delivered to every receivers through redundant paths in the mesh.

The average packet end-to-end delay is shown in Figure 4.13. The data packet latency of ANODR is increasing with nodes' mobility because when higher mobilities are introduced to the network more data packets could get lost and therefore will need to be retransmitted which increases the end-to-end delay.

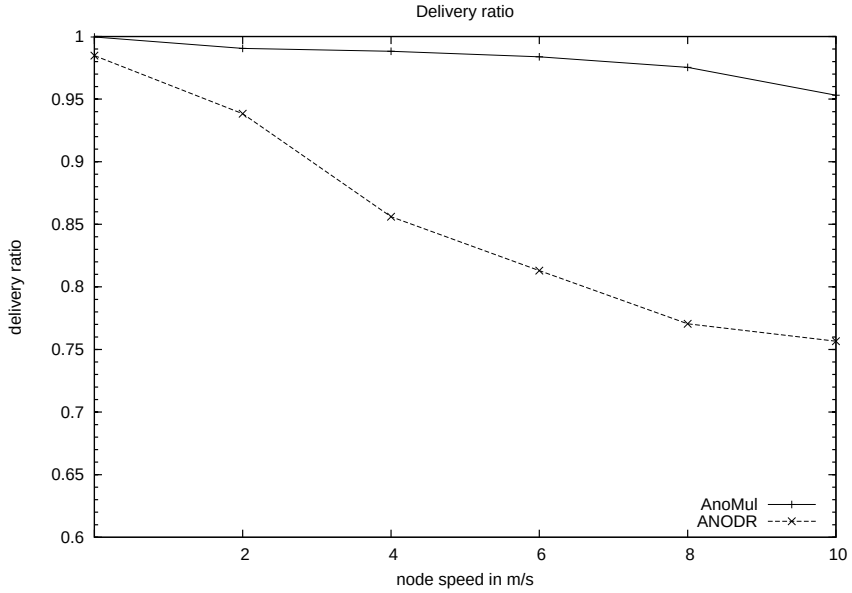

Figure 4.12: Data delivery ratio for different node mobilities for 30 receivers

Jitter is calculated as explained in section 4.6 and is shown in Figure 4.14 for a mesh with 30 group receivers for increasing node mobility. This amount of jitter is acceptable in delay sensitive applications according to the requirements given in [Adjih 2002].

As mentioned before, we define the routing overhead as the amount of routing packets in bytes divided by the sum of bytes of data packets successfully delivered to the group receivers. Figure 4.15 shows the routing overhead caused by AnoMul in the scenario of 30 receivers and compares that to the same parameter when ANODR is used to deliver data to the group receivers. To have a clear sense about the values in this graph we explain that for example a value of 0.05 in this graph means 5 bytes of routing packets is required to be transmitted in the network to make it possible to deliver 100 bytes of data packets successfully. The overhead increases with the mobility because at higher node speeds a higher number of rejoin operations will be required due to the route failures. As can be seen in this figure, in addition to the fact that ANODR cannot perform well in delivering data packets in this scenario it is even loading the network with a much higher routing overhead too.

The next set of simulations that we perform on AnoMul checks the network performance metrics for different group sizes. In this set of simulations the group size is increased from 5 to 35 and the node maximum speed is fixed at 5 m/s. Figure 4.16 shows the data packet delivery ratio when the number of group receivers is growing. As seen in this graph, the delivery ratio of AnoMul

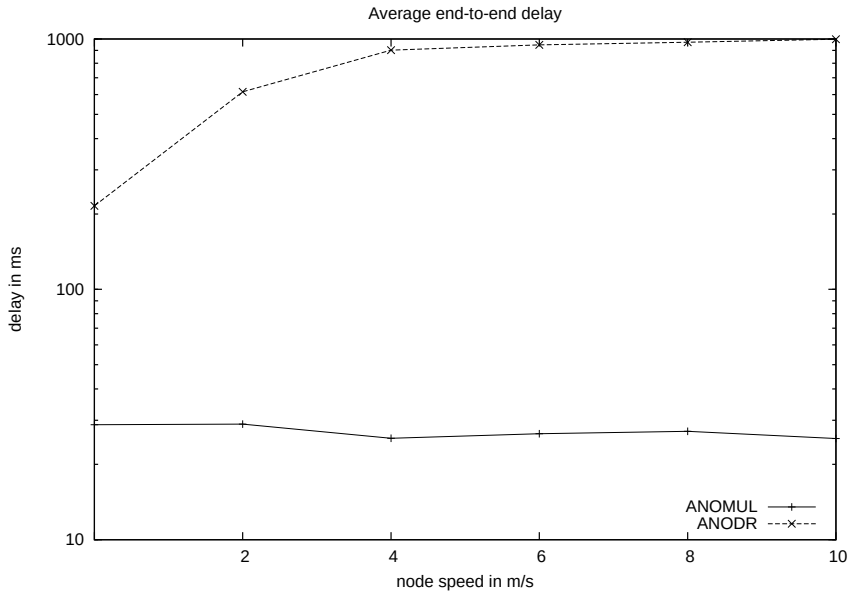

Figure 4.13: End-to-end delay for different node speeds for 30 receivers

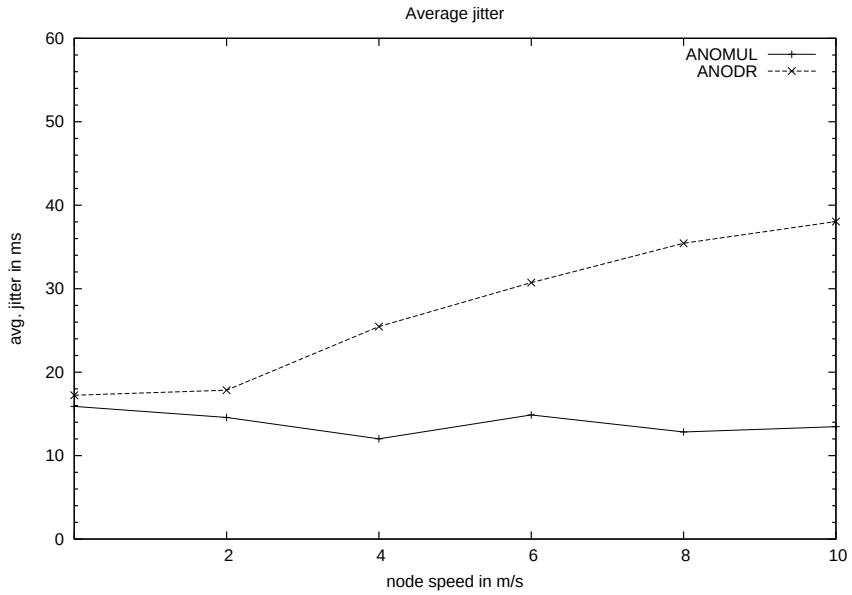

Figure 4.14: Jitter for different node speeds for 30 receivers

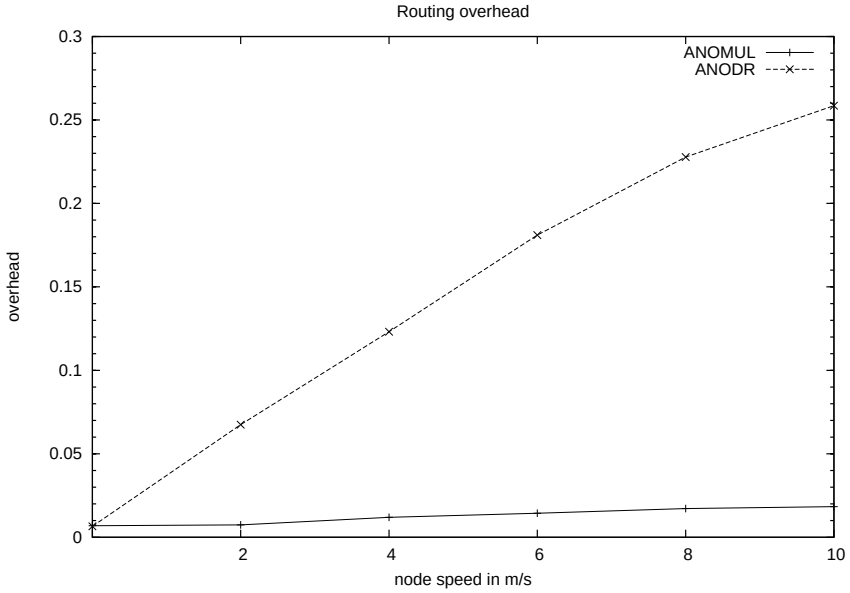

Figure 4.15: Routing overhead for different node mobilities for 30 receivers

is increasing with the number of receivers which is in contrast to the behaviour of ANODR in the same scenario. The improvement in delivery ratio of AnoMul at larger group sizes is due to the better mesh connectivity emerging at higher number of mesh members. Although this graph reports that for very small groups (when there are only low number of CBR applications which ANODR still can handle) ANODR is delivering more data packets than AnoMul, but as will be seen in next few figures regarding delay, jitter and routing overhead AnoMul is performing far better even for such scenarios.

Figure 4.17 shows the average end-to-end delay of data packets for increasing number of group receivers. As the group is growing the delay is decreasing till some point and afterward it is almost constant. This can be explained by the fact that in a richer mesh the data packets have a higher chance to find the shorter paths between *knot*s compared to a sparser group.

The jitter of the end-to-end delay in this scenario is presented in Figure 4.18. According to this graph, the jitter introduced by AnoMul increases slightly with the growth of the group size while the jitter on ANODR increases much faster.

Figure 4.19 presents the routing overhead of AnoMul for different number of group receivers. This figure indicates that as the group grows, the routing overhead normalized to the amount of data packets successfully delivered would be less. This is obviously because the required routing signalling to add and maintain a new receiver degrades as the mesh is already maintaining a high number of receivers. The new receiver can find *Knot*s close to it where the

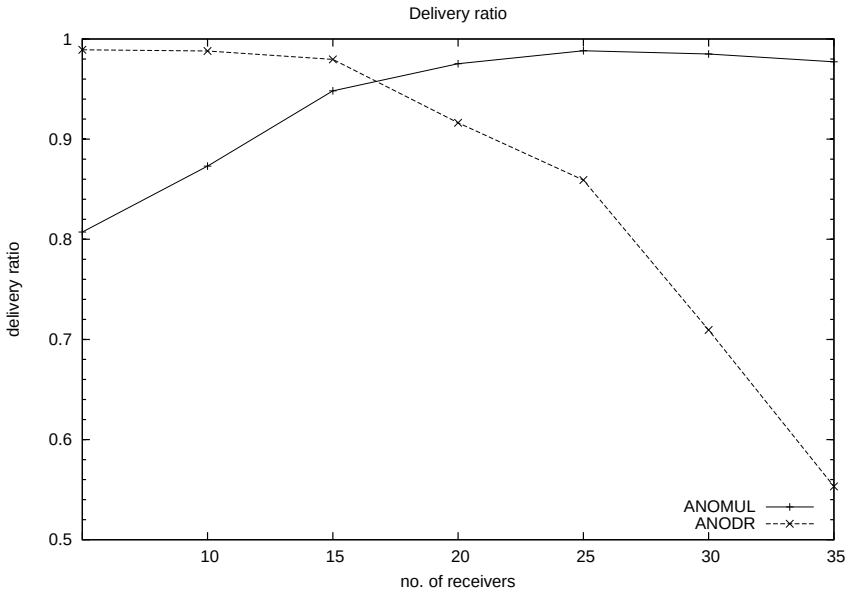

Figure 4.16: Data packet delivery ratio for growing group size

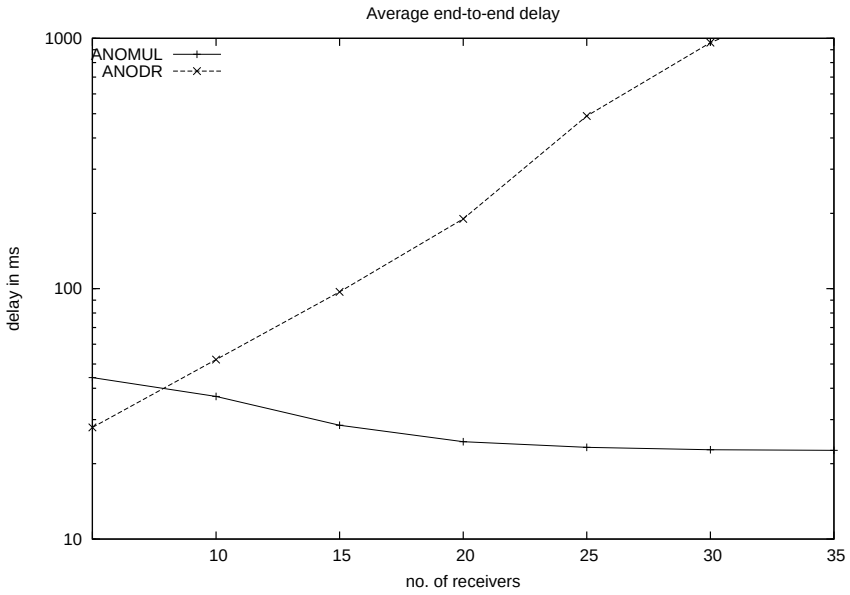

Figure 4.17: End-to-end delay for growing group size

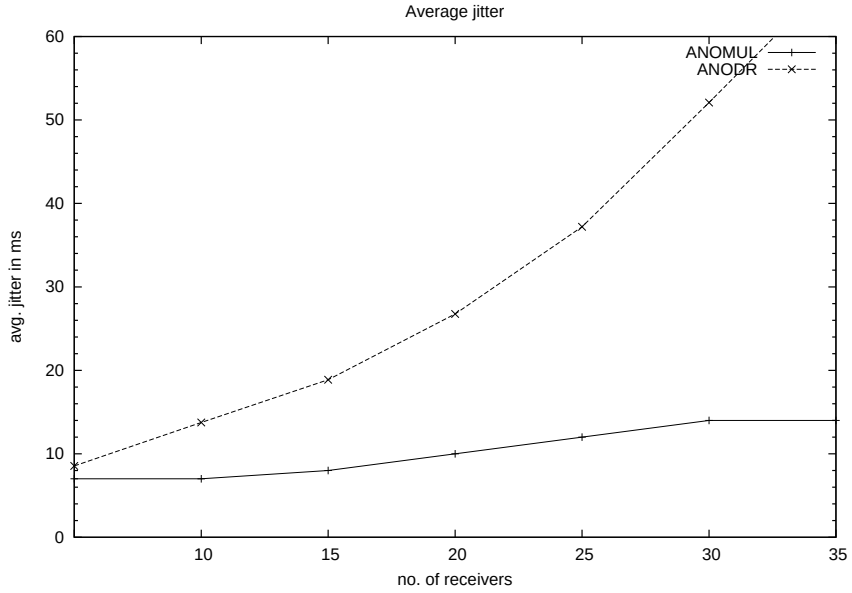

Figure 4.18: Jitter for growing group size

leader messages are already reaching.

We also investigate the effect of traffic load on AnoMul by increasing the number of source nodes sending data packets to the mesh. In this set of simulations the node speed is fixed at 5 m/s and the data rate is 1 packet per second at each sender for both AnoMul and ANODR to avoid high congestion. Figure 4.20 illustrates the data packet delivery ratio as more source nodes join the mesh. As seen in this graph, less number of data packets are delivered successfully to the receivers as there are more source nodes in the network. This is due to the packet loss as a result of the noticeably higher collision and congestion level when many sender nodes are sending their data packets to the mesh.

Figure 4.21 depicts how the amount of control overhead per delivered data byte decreases as more group sources are sending data to the mesh. This is because upon a new source joins and starts sending its packets to the mesh, the total number of data packets delivered to the receivers goes higher while almost the same amount of control packet transmissions has occurred in the whole network.

The end-to-end delay and the jitter of the latest scenario are shown in Figures 4.22 and 4.23. This graphs are reporting almost constant delay and jitter for AnoMul. As can be seen here, a very high delay and jitter is experimented when ANODR is used to deliver the data packets to the group of receivers.

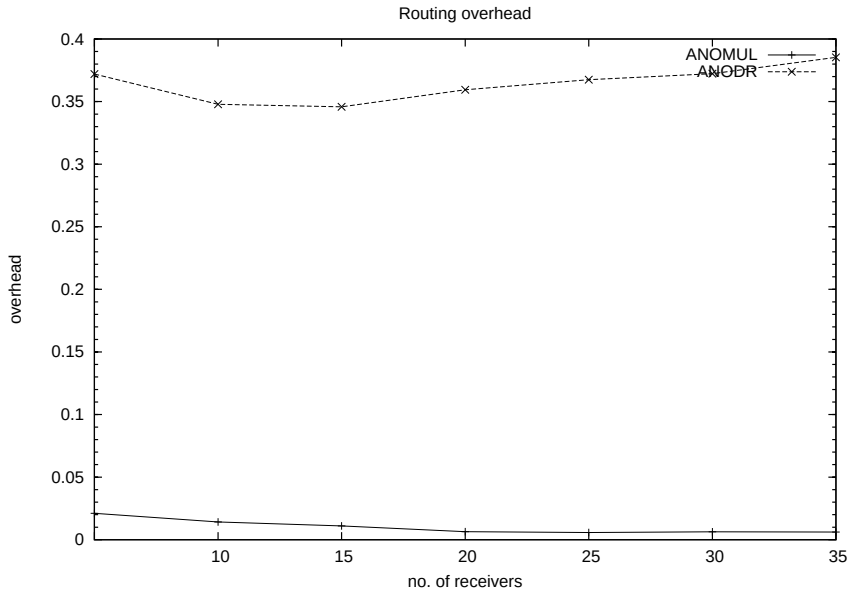

Figure 4.19: Routing overhead for growing group size

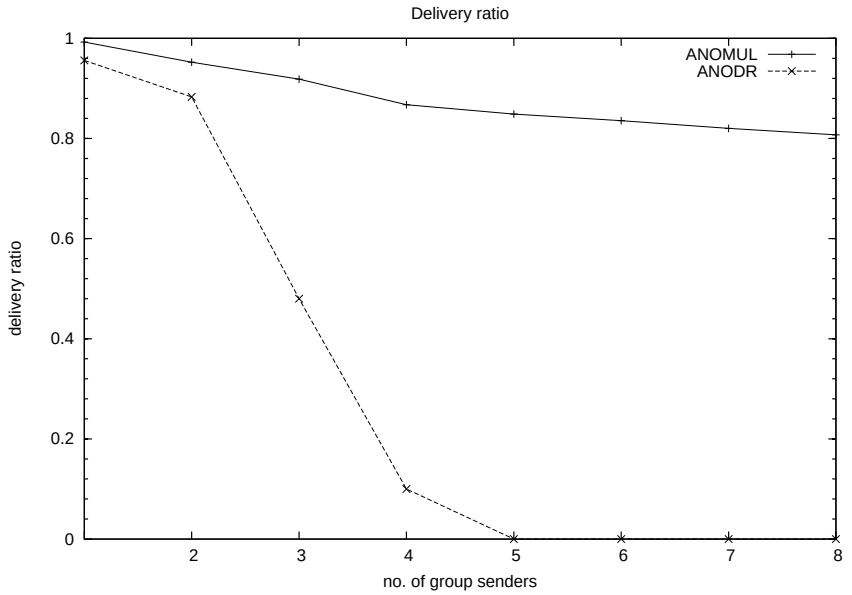

Figure 4.20: Data delivery ratio for increasing number of source nodes

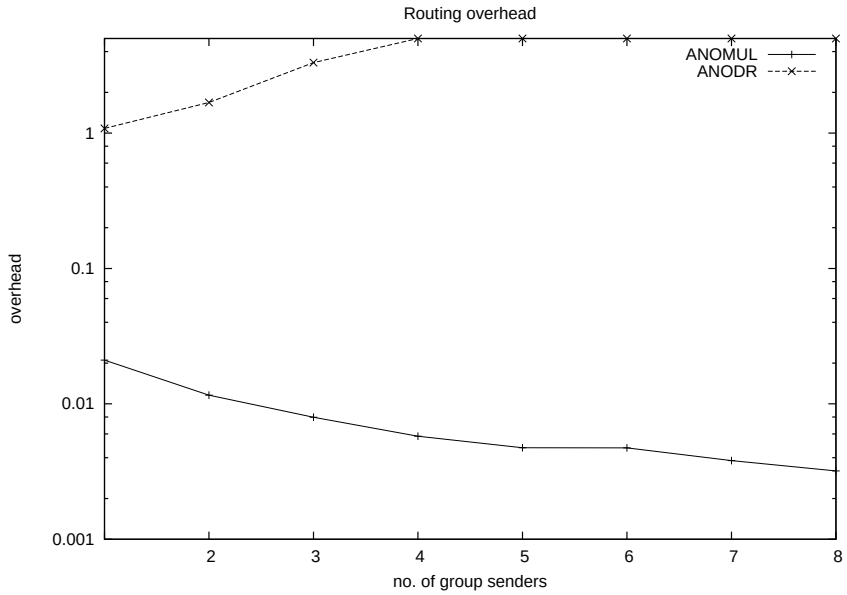

Figure 4.21: Multicast routing overhead in bytes per one byte delivered data packet for increasing number of source nodes

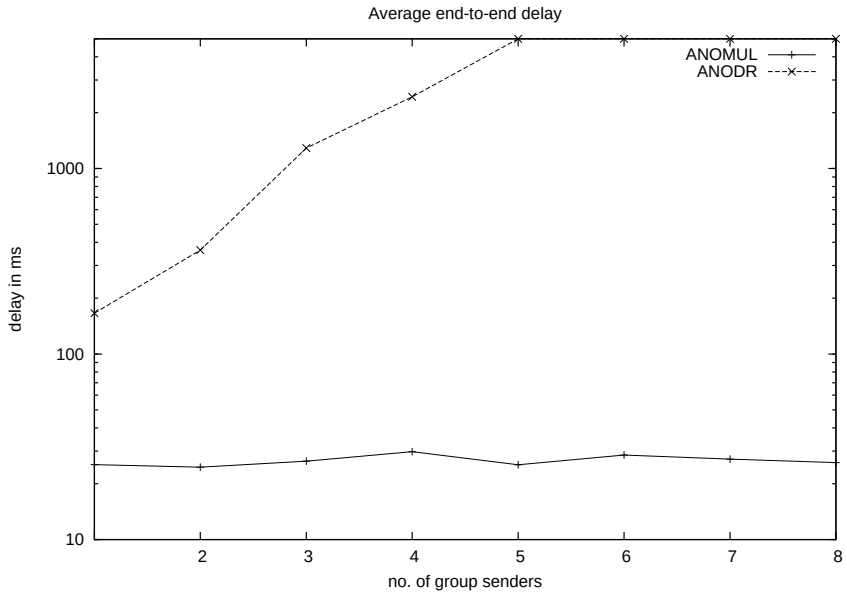

Figure 4.22: End-to-end delay for increasing number of group senders

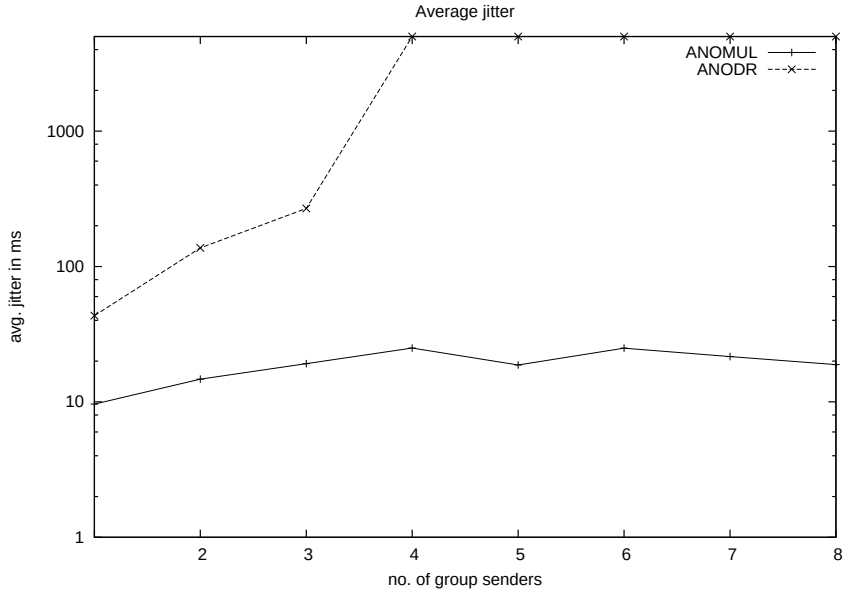

Figure 4.23: Jitter for increasing number of group senders

Discussing the results

The delay values resulted from AnoMul are in a safe range regarding delay sensitive applications of multicast ad hoc networks, as the acceptable range of end-to-end delay in voice transmission is from 150 ms to maximum 400 ms [Wang 2008].

Regarding jitter, the QoS requirements for voice and video typically require jitter to be less than 30 ms [Adjih 2002] and therefore the typical amount of jitter in AnoMul is in the proper range for those applications in which jitter is even more important than delay itself.

As mentioned before, there exist very little number of other anonymous multicast routing protocols for mobile ad hoc networks. The only similar works could be EEAMA [Kao 2007] or AMUR [Bao 2007] which investigate anonymity in multicast ad hoc networking. No performance simulation has been presented for AMUR to compare to the ones of AnoMul. The authors of AMUR just evaluated the Bloom Filter behaviour in their protocol regarding its false positive and false negative rates.

To get an idea how AnoMul is performing in comparison to similar protocols, we roughly compare the delivery ratio of AnoMul to EEAMA as follows. Based on the simulation results given in [Kao 2007], the delivery ratio of EEAMA when it is handling 6 groups each with one sender and 10 receivers and the nodes are static, is about 90%. In such a situation totally 60 multicast sender-receiver connections are handled by the network. In AnoMul according to our results in Figure 4.20 with 30 receivers and 3 senders, which means having 90 multicast

sender-receiver connections, the delivery ratio is still higher than 90%. This is while the nodes in the EEAMA scenario are static in contrast to the AnoMul nodes who are moving with the speed of 5 m/s. This means regarding congestion toleration and packet delivery ratio AnoMul is behaving very satisfying compared to EEAMA protocol.

4.7 Summary

Regarding the importance of anonymous multicast communication in ad hoc networks and regarding the fact that very few research works have been done on this area, in this chapter we proposed an anonymous multicast routing protocol for mobile ad hoc networks called AnoMul. In AnoMul we extended the identity free routing idea of ANODR protocol from unicast to multicast concerning privacy issues of the wireless group-based communication.

AnoMul is a mesh based receiver initiated multicast protocol in which we introduced some privacy mechanisms such as stepped source and the message unification technique to improve the privacy aspects of the multicast protocol. We developed a mathematical analysis as well as an implementation study to investigate the performance of the privacy mechanisms and the protocol efficiency. The analysis shows how the proposed privacy mechanisms can increase the privacy level of group senders and the mesh leader. Our simulation results present performance metrics of AnoMul including delivery ratio, delay, jitter and routing overhead and compare them to the same parameters of ANODR to show how far AnoMul outperforms ANODR in one-to-many and many-to-many anonymous communication.

Chapter 5

Conclusion and future work

In this research, we focused on anonymity and privacy issues of mobile ad hoc networks in both unicast and multicast scenarios with a special concern of location privacy.

We assumed to have a global eavesdropping adversary who is monitoring the whole network traffic to discover his favourite information about the network elements and activities. Such an adversary is the strongest possible passive adversary in wireless networks.

First we proposed a solution, called RDIS, to address destination location privacy in one-to-one communication scenarios introducing a mechanism to apply to the network layer signalling which was presented as an extension to ANODR routing protocol. However, this idea could be used in an appropriate way for some other ad hoc routing protocols as well.

When RDIS protocol is used, the destination node could not be distinguished by the eavesdropping adversary among an anonymity set of nodes depending on the protocol parameters. Not only the route reply packet flow is hidden among the route discovery signalling, but the data packets are routed on a ring formed by two routes to make it impossible for the adversary to find the destination of the communication. Using cloud idea, route privacy could be achieved too. Privacy analysis and simulation study were presented to evaluate the protocol. The simulation results indicate that RDIS can guaranty a good privacy level for the receiver while the performance of the network is maintained.

In the second part of this dissertation, we designed a multicast protocol based on an underlying unicast routing infrastructure to provide anonymity and location privacy for the elements of multicast communication. This protocol, called AnoMul, is a mesh based multicast protocol designed in network layer for MANETs in which the mesh is initiated by the group receivers. Every group member can hide its identification from the rest of the network and from the outsiders. The group senders can send their data packets to the group anonymously while their location is also protected from the adversary. The location privacy of the leader of the group is also protected. The privacy analysis and the simulation results show that the privacy goals and a good performance are both achieved by AnoMul protocol.

In our future work, we are interested to apply the ideas of RDIS on other routing protocols to compare the achievements. Also we may investigate designing a more efficient mechanism for achieving route privacy in MANET unicast

communication.

In AnoMul we did not consider the cases in which more than one leader emerges in the network and sub-mesh group communication is formed. Such sub-groups could be unified by designing proper mechanisms to end up to a single mesh with one leader all the time. This could be another consideration of future work.

In the current version of AnoMul it is possible for a global eavesdropper to find the location of group receivers when they are joining the mesh by tracing their joining signalling. Another plan for our future work could focus on location privacy of the group receivers.

Bibliography

[Adjih 2002] C. Adjih, T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum and L. Viennot. *Cisco AVVID Network Infrastructure Enterprise Quality of Service Design.* Reference Network Design, August, 2002. (Cited on pages 99 and 106.)

[Amuthan 2011] A. Amuthan and D. Nagamani Abirami. *MULTICAST SECURITY ATTACKS AND ITS COUNTER MEASURES FOR PUMA PROTOCOL.* International Journal of Computer Technology and Applications, 2011. (Cited on page 20.)

[ARGYROUDIS 2005] PATROKLOS G. ARGYROUDIS and DONAL O'MAHONY. *SECURE ROUTING FOR MOBILE AD HOC NETWORKS.* IEEE communications surveys, Third quarter 2005. (Cited on page 24.)

[Bao 2007] L. Bao. *A new approach to anonymous multicast routing in Ad Hoc Networks.* In Proceedings of the Second International Conference on Communications and Networking in China (CHINACOM), 2007. (Cited on pages 76 and 106.)

[Boukerche 2004] Azzedine Boukerche, Khalil El-Khatib, Li Xu and Larry Korba. *SDAR: A Secure Distributed Anonymous Routing Protocol for Wireless and Mobile Ad Hoc Networks.* In LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, pages 618–624, Washington, DC, USA, 2004. IEEE Computer Society. (Cited on pages 35, 36 and 37.)

[Boukerche 2006] Azzedine Boukerche, Khalil El-khatib, Li Xu and Larry Korba. *Performance evaluation of an anonymity providing protocol for wireless ad hoc networks.* Performance Evaluation, vol. 63, pages 1094–1109, 2006. (Cited on page 27.)

[C 2009] Edith C and H. Ngai. *On providing sink anonymity for sensor networks.* In International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, Leipzig, Germany, 2009. (Cited on page 51.)

[Camenisch 1997] J. Camenisch and M. Stadler. *Efficient group signature schemes for large groups.* In Advances in Cryptology CRYPTO'97, vol. 1296 of LNCS, pages 410–424. Springer-Verlag, 1997. (Cited on page 79.)

[Camenisch 1998] J. Camenisch and M. Michels. *A group signature scheme based on an RSA-variant.* Rapport technique, University of Aarhus, 1998. (Cited on page 79.)

[Cayirci 2009] Erdal Cayirci and Chunming Rong. Security in wireless ad hoc and sensor networks. Wiley, 2009. (Cited on pages 6, 19, 20, 22, 25 and 26.)

[Chatzikokolakis 2007] Konstantinos Chatzikokolakis. *Probabilistic and Information-Theoretic Approaches to Anonymity.* PhD thesis, Ecole Polytechnique of Paris, October 2007. (Cited on page 28.)

[Chaum 1981a] David L. Chaum. *Untraceable electronic mail, return addresses, and digital pseudonyms.* Commun. ACM, vol. 24, no. 2, pages 84–90, 1981. (Cited on pages 27 and 49.)

[Chaum 1981b] David L. Chaum. *Untraceable electronic mail, return addresses, and digital pseudonyms.* Commun. ACM, vol. 24, no. 2, pages 84–90, 1981. (Cited on pages 32 and 42.)

[Chaum 1988] David Chaum. *The dining cryptographers problem: Unconditional sender and recipient untraceability.* Journal of Cryptology, vol. 1, 1988. (Cited on pages 27 and 49.)

[Claudia Diaz 2002] Joris Claessens Claudia Diaz Stefaan Seys and Bart Preneel. *Towards measuring anonymity.* In Proceedings of Privacy Enhancing Technologies Workshop (PET 2002). Springer-Verlag, LNCS 2482, April 2002. (Cited on pages 42 and 43.)

[Compton] Stuart Compton. *802.11 Denial of Service Attacks and Mitigation.* In featured in the SANS Reading Room. (Cited on page 15.)

[Dingledine 2004] Roger Dingledine, Nick Mathewson and Paul Syverson. *Tor: The Second-Generation Onion Router.* In Proceedings of the 13th USENIX Security Symposium, pages 303–320, 2004. (Cited on page 30.)

[Dirk Balfanz 2003] Narendar Shankar Diana K. Smetters Jessica Staddon Dirk Balfanz Glenn Durfee and Hao-Chi Wong. *Secret Handshakes from Pairing-Based Key Agreements.* In Proceedings of the 2003 IEEE Symposium on Security and Privacy (S & P 2003), pages 180–196, 11-14 May 2003. (Cited on page 38.)

[Douglas J. Kelly 2008] Richard A. Raines Michael R. Grimaila Douglas J. Kelly Rusty O. Baldwin and Barry E. Mullins. *A Survey of State-of-the-Art in Anonymity Metrics.* In NDA'08. Fairfax, Virginia, USA, October 2008. (Cited on page 42.)

[El-Khatib 2003] K. El-Khatib, L. Korba, R. Song and G. Yee. *Secure Dynamic Distributed Routing Algorithm for Ad Hoc Wireless Networks.* In ICPP Workshops, 2003. (Cited on page 75.)

[Fonseca 2006] Emanuel Fonseca and Andreas Festag. *A Survey of Existing Approaches for Secure Ad Hoc Routing and Their Applicability to VANETS.* Rapport technique, NEC Network Laboratories, 2006. (Cited on page 25.)

[Garcia-Luna-Aceves 999] J. J. Garcia-Luna-Aceves and E. L. Madruga. *The Core Assisted Mesh Protocol.* In IEEE Journal on Selected Areas in Communications, Special Issue on Ad-Hoc Networks, 17:1380-1394 (1999). (Cited on page 72.)

[Gergely Tóth 2004] Zoltán Hornák Gergely Tóth and Ferenc Vajda. *Measuring anonymity revisited.* In In Sanna Liimatainen and Teemupekka Virtanen, editors, Proceedings of the Ninth Nordic Workshop on Secure IT Systems, pages 85–90. Finland, November 2004. (Cited on page 43.)

[Goldschlag 1999] David Goldschlag, Michael Reed and Paul Syverson. *Onion Routing for Anonymous and Private Internet Connections.* Communications of the ACM, vol. 42, pages 39–41, 1999. (Cited on page 29.)

[Gould 1855] B. A. Gould. *On peirce's criterion for the rejection of doubtful observations, with tables for facilitating its application.* In Astronomical Journal, volume IV, pages 81–87, Apr. 1855. (Cited on page 44.)

[He 2004] Qi He, Dapeng Wu and Pradeep Khosla. *Quest for Personal Control over Mobile Location Privacy.* IEEE Communications Magazine, vol. 42, 2004. (Cited on page 48.)

[Hong 2006] Xiaoyan Hong, Jiejun Kong and Mario Gerla. *Mobility changes anonymity: new passive threats in mobile ad hoc networks: Research Articles.* volume 6, pages 281–293, Chichester, UK, 2006. John Wiley and Sons Ltd. (Cited on page 50.)

[Hu 2003] Yih-Chun Hu, David B. Johnson and Adrian Perrig. *SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks.* In Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications WMCSA '02, 2003. (Cited on page 23.)

[Johnson 2001] David B. Johnson, David A. Maltz and Josh Broch. *DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks.* In C.E. Perkins, editeur, Ad Hoc Networking, chapitre 5, pages 139–172. Addison-Wesley, 2001. (Cited on page 21.)

[ju Lee 1999] Sung ju Lee, William Su and Mario Gerla. *On-Demand Multicast Routing Protocol*. In Proceeding of WCNC, pages 1298–1302, 1999. (Cited on page 72.)

[Kamat 2005] Pandurang Kamat, Yanyong Zhang, Wade Trappe and Celal Ozturk. *Enhancing Source-Location Privacy in Sensor Network Routing*. In ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, pages 599–608, Washington, DC, USA, 2005. IEEE Computer Society. (Cited on pages 48 and 50.)

[Kao 2007] Jung-Chun Kao and Radu Marculescu. *Energy-efficient anonymous multicast in mobile ad-hoc networks*. In Proceedings of ICPADS 2007, 2007. (Cited on pages 76 and 106.)

[Kong 2003] Jiejun Kong and Xiaoyan Hong. *ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks*. In MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, pages 291–302, New York, NY, USA, 2003. (Cited on pages 10, 33, 35, 48, 50, 71, 75, 77, 78 and 89.)

[Kong 2004] Jiejun Kong. *Anonymous and untraceable communications in mobile wireless networks*. PhD thesis, 2004. Chair-Gerla, Mario. (Cited on page 56.)

[Kong 2007] Jiejun Kong and Xiaoyan Hong. *An Identity-Free and On-Demand Routing Scheme against Anonymity Threats in Mobile Ad Hoc Networks*. IEEE Transactions on Mobile Computing, vol. 6, no. 8, pages 888–902, 2007. (Cited on pages 31, 32, 33, 34, 35 and 48.)

[Kunz 2004] Thomas Kunz. *Multicast Versus Broadcast in a MANET*. In ADHOC-NOW, pages 14–27, 2004. (Cited on pages 9 and 71.)

[M 2011] Rajendiran. M and Srivatsa. S. K. *On-Demand Multicasting in Adhoc Networks: Performance Evaluation of AODV, ODMRP and FSR*. In IJCSI International Journal of Computer Science Issues, volume 8, May 2011. (Cited on page 74.)

[Ma 2011] Zhendong Ma. *Location Privacy in Vehicular Communication Systems: a Measurement Approach*. PhD thesis, Ulm university-Germany, 2011. (Cited on pages 42 and 46.)

[Makki 2007] S. Kami Makki, Peter Reiher, Kia Makki, Niki Pissinou and Shamila Makki. Mobile and wireless network security and privacy. Springer, 2007. (Cited on pages 32, 33, 34, 36, 37 and 38.)

[Mamatha 2010] G.S. Mamatha and Dr. S.C. Sharma. *Article:Network Layer Attacks and Defense Mechanisms in MANETS- A Survey.* International Journal of Computer Applications, vol. 9, no. 9, pages 12–17, November 2010. Published By Foundation of Computer Science. (Cited on pages 13 and 14.)

[Martucci 2009] Leonardo A. Martucci. *Identity and Anonymity in Ad Hoc Networks.* PhD thesis, Karlstad University, 2009. (Cited on pages 17, 20, 22, 31, 38 and 39.)

[Md. Saiful Azad 2009] Md. Arafatur Rahman Aisha H. Abdalla Akhmad Unggul Priantoro Md. Saiful Azad Farhat Anwar and Omer Mahmoud. *Performance Comparison of Proactive and Reactive Multicast Routing Protocols over Wireless Mesh Networks.* In IJCSNS International Journal of Computer Science and Network Security, volume 9, June 2009. (Cited on page 74.)

[Mehta 2007] K. Mehta, Donggang Liu and Matthew Wright. *Location Privacy in Sensor Networks Against a Global Eavesdropper.* In ICNP, pages 314–323. IEEE, 2007. (Cited on page 51.)

[Merwe 2007] Johann Van Der Merwe, Dawoud Dawoud and Stephen McDonald. *A survey on peer-to-peer key management for mobile ad hoc networks.* ACM Comput. Surv., vol. 39, no. 1, page 1, 2007. (Cited on page 4.)

[Michel Barbeau 2006] Jeyanthi Hall Michel Barbeau and Evangelos Kranakis. *Detecting Impersonation Attacks in Future Wireless and Mobile Networks.* Lecture Notes in Computer Science, 2006. (Cited on page 19.)

[Neeraj Jaggi 2011] Umesh MarappaReddy Neeraj Jaggi and Rajiv Bagai. *A Three-Dimensional Approach Towards Measuring Sender Anonymity.* In The First International Workshop on Security in Computers, Networking and Communications, 2011. (Cited on pages 43 and 44.)

[Nguyen 2008] Hoang Lan Nguyen and Uyen Trang Nguyen. *A study of different types of attacks on multicast in mobile ad hoc networks.* Ad Hoc Netw., vol. 6, no. 1, pages 32–46, 2008. (Cited on pages 18 and 20.)

[N.Shanthi 5 09] Dr. Lganesan N.Shanthi and Dr. K. Ramar. *Study of different attacks on multicast mobile ad hoc networks.* Journal of Theoretical and Applied Information Technology, 2005-09. (Cited on page 20.)

[Peirce 1852] B. Peirce. *Criterion for the rejection of doubtful observations.* In Astronomical Journal, volume II, pages 161–163, 1852. (Cited on page 44.)

[Perkins 1999] Charles E. Perkins and Elizabeth M. Royer. *Ad-hoc On-Demand Distance Vector Routing.* wmcsa'99, pages 90–100, 1999. (Cited on page 21.)

[Pfitzmann 2008] Andreas Pfitzmann and Marit Hansen. *Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology*, February 2008. (Cited on pages 2 and 3.)

[Ping Yi 2011] Yan Zou Ping Yi Futai Zou and Zhiyang Wang. *Performance analysis of mobile ad hoc networks under flooding attacks.* Journal of Systems Engineering and Electronics, 2011. (Cited on page 18.)

[Pradip M. Jawandhiya 2010] DR. M.S. Ali Pradip M. Jawandhiya Mangesh M. Ghonge and J.S. Deshpande. *A Survey of Mobile Ad Hoc Network Attacks.* International Journal of Engineering Science and Technology, 2010. (Cited on pages 17 and 18.)

[Qua] *Scalable Network Technologies (SNT). http://www.qualnet.com.* (Cited on pages 64 and 97.)

[Reiter 1998] Michael K. Reiter and Aviel D. Rubin. *Crowds: anonymity for Web transactions.* ACM Transactions on Information and System Security, vol. 1, no. 1, pages 66–92, 1998. (Cited on pages 28, 29, 41 and 49.)

[Rescorla 1999] E. Rescorla. *Diffie-Hellman Key Agreement Method, RFC 2631, Internet Engineering Task Force, 1999.*, June 1999. (Cited on page 77.)

[RFC] *RFC 1889: RTP: A Transport Protocol for Real-Time Applications http://www.ietf.org/rfc/rfc1889.txt.* (Cited on page 97.)

[Royer 1999] Elizabeth M. Royer and Charles E. Perkins. *Multicast operation of the ad-hoc on-demand distance vector routing protocol.* In Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom '99, pages 207–218, New York, NY, USA, 1999. ACM. (Cited on page 72.)

[Sanzgiri 2005] Kimaya Sanzgiri, Daniel Laflamme, Bridget Dahill, Brian Neil, Levine Clay, Shields Elizabeth and M. Belding-royer. *Authenticated routing for ad hoc networks.* IEEE Journal On Selected Areas In Communications, vol. 23, pages 598–610, 2005. (Cited on page 24.)

[Serjantov 2002] Andrei Serjantov and George Danezis. *Towards an information theoretic metric for anonymity.* April 2002. (Cited on page 42.)

[Seys 2006] Stefaan Seys and Bart Preneel. *ARM: Anonymous Routing Protocol for Mobile Ad hoc Networks.* pages 133–137, 2006. (Cited on pages 39, 49, 50 and 53.)

[Sirer] Emin GÃijn Sirer, Milo Polte and Mark Robson. *CliqueNet: A Self-Organizing, Scalable, Peer-to-Peer Anonymous Communication Substrate*. (Cited on page 27.)

[Sung-Ju Lee 2000] Julian Hsu Mario Gerla Sung-Ju Lee William Su and Rajive Bagrodia. *A performance comparison study of ad hoc wireless multicast protocols*. In INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Israel, 2000. (Cited on page 74.)

[Sweeney 2002] L. Sweeney. *k-anonymity: a model for protecting privacy*. In International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, volume 10, pages 557–570, 2002. (Cited on page 45.)

[Sy 2006] Denh Sy, Rex Chen and Lichun Bao. *ODAR: On-Demand Anonymous Routing in Ad Hoc Networks*. In Proceedings of The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), pages 267–276, 2006. (Cited on pages 39 and 76.)

[Taheri] S. Taheri, S. Hartung and D. Hogrefe. *RDIS: Destination Location Privacy in MANETs*. To appear in the International Journal of Information Privacy, Security and Integrity. (Cited on page 48.)

[Taheri 2009] Somayeh Taheri and Dieter Hogrefe. *An Anonymous Multicast Routing Protocol For Mobile Ad Hoc Networks*. In 16th ACM Conference on Computer and Communications Security CCS'09, Poster session, 9-13 November 2009. (Cited on page 71.)

[Taheri 2010] S. Taheri, S. Hartung and D. Hogrefe. *Achieving receiver location privacy in Mobile Ad Hoc Networks*. In Proceedings of IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT2010), 2010. (Cited on pages 48 and 75.)

[Tuomas Aura 2008] Michael Roe Tuomas Aura Janne Lindqvist and Anish Mohammed. *Chattering laptops*. Proceedings of the 8th International Symposium on Privacy Enhancing Technologies (PETS 2008), 2008. (Cited on page 23.)

[Vaishampayan 2004] R. Vaishampayan and J.J. Garcia-Luna-Aceves. *Efficient and Robust Multicast Routing in Mobile Ad Hoc Networks*. In Proceedings of the IEEE international conference on mobile ad-hoc and sensor systems, pages 304–313, 2004. (Cited on page 72.)

[Wang 2008] Ping Wang. *Distributed Medium Access control for QoS support in Wireless Networks*. PhD thesis, University of Waterloo, 2008. (Cited on pages 98 and 106.)

[Westin 1967] Alan Westin. Privacy and freedom. New York, 1967. (Cited on page 2.)

[wik] *Wikipedia: http://en.wikipedia.org/wiki/Privacy.* (Cited on page 2.)

[Wu 2008] Xiaoxin Wu, Jun Liu, Xiaoyan Hong and Elisa Bertino. *Anonymous Geo-Forwarding in MANETs through Location Cloaking.* volume 19, pages 1297–1309, Piscataway, NJ, USA, 2008. IEEE Press. (Cited on page 51.)

[Xiao 2006] Li Xiao, Xiaomei Liu, Wenjun Gu, Dong Xuan and Yunhao Liu. *A design of overlay anonymous multicast protocol.* In Proceedings of the 20th international conference on Parallel and distributed processing, IPDPS'06, pages 48–48, Washington, DC, USA, 2006. IEEE Computer Society. (Cited on page 75.)

[Xiaodong Lin 2008] Chenxi Zhang Haojin Zhu Pin-Han Ho Xiaodong Lin Rongxing Lu and Xuemin Shen. *Security in Vehicular Ad Hoc Networks.* IEEE Communications Magazine, vol. 46, no. 4, pages 88–95, Apr 2008. (Cited on page 6.)

[Xie 2002] Jason Xie, Rajesh R. Talpade, Anthony McAuley and Mingyan Liu. *AMRoute: Ad Hoc Multicast Routing Protocol.* MONET, vol. 7, no. 6, pages 429–439, 2002. (Cited on page 72.)

[Y. Xiao 2006] X. Shen Y. Xiao and D.-Z. Du (Eds.). A survey on attacks and countermeasures in mobile ad hoc networks, wireless/mobile network security. Springer, 2006. (Cited on page 18.)

[Yang 2008] Yi Yang, Min Shao, Sencun Zhu, Bhuvan Urgaonkar and Guohong Cao. *Towards event source unobservability with minimum network traffic in sensor networks,.* In The ACM Conference on Wireless Network Security (WiSec), 2008. (Cited on page 51.)

[Y.C. Hu 2005] H.J. Wang Y.C. Hu. *A framework for location privacy in wireless networks.* In ACM SIGCOMM Asia Workshop 2005, 2005. (Cited on pages i, 8, 48 and 52.)

[Yi 2002] Seung Yi, Prasad Naldurg and Robin Kravets. *A Security-Aware Routing Protocol for Wireless Ad Hoc Networks.* In Processing of ACM symposium on mobile ad hoc networking and computing (MOBIHOC), pages 286–292, 2002. (Cited on page 25.)

[Yuxin Deng 2007] Jun Pang Yuxin Deng and Peng Wu. *Measuring anonymity with relative entropy.* In Proceedings of the 4th International Workshop on Formal Aspects in Security and Trust, volume 4691 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2007. (Cited on page 45.)

[Zapata 2002] G. M. Zapata. *Secure Ad hoc On-Demand Distance Vector Routing.* In PROCEEDINGS OF ACM Mobile Computing and Communications Review (MC2R), pages 6(3):106–107, July 2002. (Cited on page 25.)

[Zhang 2005] Yanchao Zhang, Wei Liu and Wenjing Lou. *Anonymous communications in mobile ad hoc networks.* In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, volume 3, pages 1940–1951 vol. 3, 2005. (Cited on pages 50 and 75.)

[Zhu 2004] B. Zhu, Z. Wan, M. S. Kankanhalli, F. Bao and R. H. Deng. *Anonymous Secure Routing in Mobile Ad-Hoc Networks.* In IEEE LCN, 2004. (Cited on page 75.)