# Reliable and Secure Geocasting in VANETs

by

Antonio Prado Bernia

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

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

# Abstract

Current geocasting algorithms for VANETs are being designed to enable either private or reliable communications, but not both. Existing algorithms preserve privacy by minimizing the information used for routing, and sacrifice message delivery success. On the other hand, reliable protocols often store node information that can be used to compromise a vehicle's privacy. We have designed two private and reliable geocasting protocols for VANETs that ensure confidentiality. One is a probabilistic algorithm that uses direction-based dissemination, while the other is a deterministic algorithm that uses transmission-coverage dissemination. To preserve privacy, we create unlinkable and pseudonymous channels of communication with geocasting. For encryption and authentication, we use a public key technique. Our probabilistic forwarding model depends on message rate and cumulative payload, as well as the value of the angle of spreading of the direction-based scheme. To reduce message duplication, we apply dynamic traffic restriction and probabilistic forwarding techniques. The deterministic forwarding algorithm delays forwarding messages based on its uncovered transmission area after neighbouring nodes have broadcast the message. We prove that both algorithms ensure node privacy with appropriate message encryption security, and we ran simulations to demonstrate that both meet the message delivery requirements. From the gathered data, we observe that both algorithms behave differently depending on the scenario, with node density affecting the deterministic algorithm, while the angle of spreading does have a significant impact on the probabilistic protocol.

**Acknowledgements**

I would like to thank everyone who made this thesis possible by providing their ideas, advice and support. My most sincere thanks go to my supervisors, Professor Amiya Nayak and Professor Ivan Stojmenovic. Special thanks as well to Dr. Sushmita Ruj, whose guidance helped me complete this thesis.

I would also like to thank my family and friends, as well as my wife, who encouraged me to obtain my master's degree and helped me with the thesis-writing process. I cannot overstate how valuable your support was.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 VANETs

In our society, vehicles are used regularly to transport people and resources over large distances quickly and conveniently. While the ease of use and access to vehicles make this method of transportation appealing, there are also inherent risks to driving. The high speeds, combined with the large number of vehicles, makes driving-related accidents an unfortunately all-too-common occurrence: even a short distraction (Ranney et al., 2000) can prove to be fatal. According to a National Highway Traffic Safety Administration (NHTSA) report, in the year 2011, a total of 32,310 people died as a result of a traffic accident (Nat, 2012). In Canada, the Canadian Council of Motor Transport Administrators (CCMTA) reports that in 2009 there were 123,192 personal-injury collisions, with 2209 fatalities and 11,451 injured as an outcome (Can, 2011). Some of these accidents could have been avoided if the driver had received some early warning, such as an up-to-date road condition report, or information regarding traffic-flow diversion (i.e. to move aside when an emergency response vehicle needs priority access to a lane).

Hence, the concept of *Vehicular Ad Hoc Networks* (VANETs) was developed, in which vehicles are equipped with sensors to record and analyze their environment as well as with communication devices that allow the exchange of information with other vehicles. Vehicles broadcast and receive messages, and through an interface – such as a warning bell, a screen, or even a light on the dashboard – drivers are made aware of road conditions in their surroundings, and as a result can react to events on the road more appropriately to avoid an accident. For example, drivers on a highway could be warned of icy conditions before they on even reach an area, giving them the chance to slow down the vehicle to better negotiate the road. Moreover, if a vehicle disseminates a message on congestion in its communication range, it can help other vehicles choose alternative routes and in doing so congestion can be somewhat reduced.

Looking into the future, apart from traffic safety, we could also use VANETs for commodity systems and applications, such as automated toll payment, in-vehicle Internet access, information on local places of interest, and even on-demand multimedia entertainment.

These so-called Intelligent Transportation Systems (ITS) combine communications technology and information applications in a vehicle to provide its passengers with a superior driving experience.

### 1.1.1 Architecture of VANETs

The nodes in a VANET are not required to be homogenized, which is to say that the network is not restricted only to vehicles on the road. Different types of nodes provide different capabilities to the environment, due to their nature and abilities.

Road side units (RSUs) are devices that have limited (or no) mobility and can communicate with the VANET. They can be used as gateways between the vehicles in the

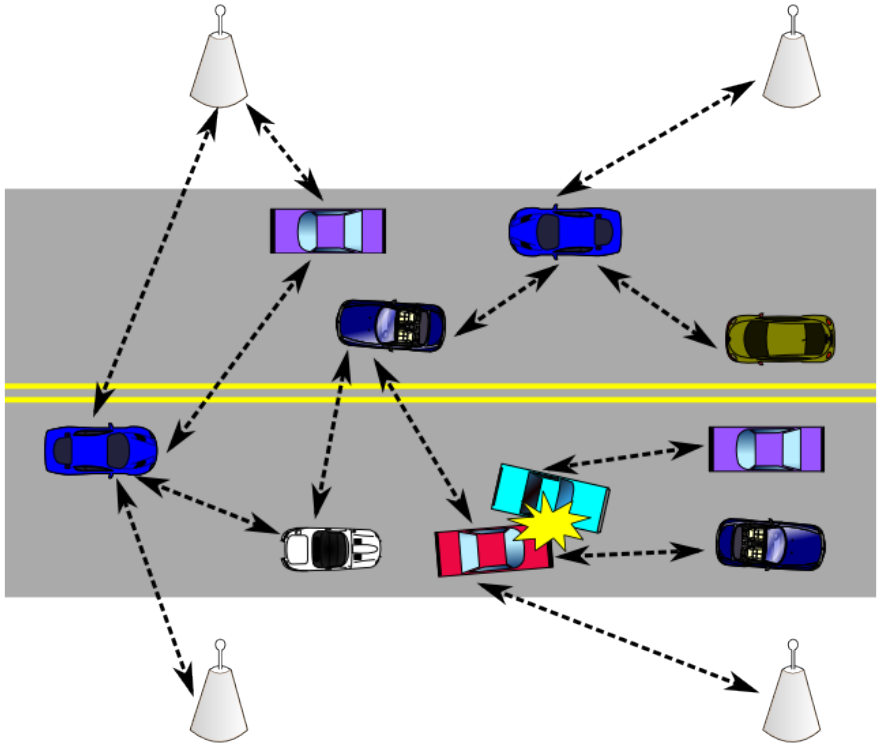

Figure 1.1: Architecture of VANETs.
Vehicles can communicate with each other and with Road Side Units (RSUs).

network and the world at large, as well as providers of services that are tied to the area. For example, in the case of road construction work, a RSU could be set up temporarily in the construction area to indicate to drivers to change lanes, both visually – by way of a street sign – and electronically – by broadcasting a message to that effect. On the other hand, existing networks, such as cellular networks or even the Internet, could interact with VANETs via specialized RSUs which could smartly route messages between both networks.

VANETs that want to employ encryption models often make use of trusted third parties, such as certificate authorities (CA), which are agreed-upon entities that offer services for generating, revoking, managing and validating node identification, by way of digital certificates. With digital certificates, a vehicle that receives a message can

confirm that the message sender is who it claims to be, and not an imposter in the network. Moreover, with signed and encrypted messages, a sender ensures that such an imposter cannot modify its messages without being detected.

Finally, the vehicles themselves may have a wide range of sensors and abilities. Undoubtedly, as the technology improves, so will the accuracy of the sensors installed in a vehicle, as well as the variety of inputs to monitor, such as temperature, velocity, luminosity, etc. Even when a vehicle does not have advanced sensor abilities, it could still receive information obtained from other better equipped vehicles.

### 1.1.2 Difference Between VANETs and MANETs

At first glance, a VANET can be classified as a specific implementation of a Mobile Ad-hoc Network MANET (Parno and Perrig, 2005). The argument for this is based on the fact that both systems use the same physical communication layer (802.11p) and are designed to have mobile and fixed nodes in the network. There has been an extensive amount of research done on MANETs regarding routing, security and efficiency, and it has been thought that this research can be applied directly (or with small modifications) to VANETs. However, while the underlying concept is very similar, VANETs and MANETs also have differing characteristics, such as:

- Energy: For many MANETs, the conservation of energy, or at the very least, the efficient management of it, is critical for the design features of the network. Nodes in the network have a limited amount (or supply) of energy, and message broadcasting decisions must take this factor into consideration. In some scenarios, such as sensor networks, message transmission and reception use more energy than what is required for the node to sense and record data (Feeney and Nilsson, 2001; Feeney, 2004). Moreover, increasing the transmission range comes at the cost of energy.

In those scenarios, a node may choose to send a message a small distance away, or even refuse to forward a message, to record more information.

In contrast, the vehicles and RSUs of a VANET have a near-limitless supply of energy, by virtue of vehicles being refueled regularly and RSUs being connected to the electric grid. While it is still advisable to restrict the range of transmission, it is to minimize interference with other transmissions.

- VANETs are ephemeral networks (Raya and Hubaux, 2005): Nodes in a MANET can be mobile, yet the velocity of any given node is still relatively small compared to a vehicle moving on a road. The range of velocity and/or acceleration is also greater in a VANET, which makes it more difficult to predict the position of a given node during a time period, and forces communications between vehicles to be near instantaneous.

  This movement also causes the topology of the network to be volatile in a VANET. Therefore, algorithms that use network topology to efficiently route messages in a MANET are not as efficient in a VANET (Abuelela et al., 2008).

  That being said, vehicles in a VANET move along very well defined paths, such as highways, roads and streets, while nodes in a MANET have a larger degree of freedom of movement.

- Network size: A VANET has a larger number of nodes (vehicles) than a MANET. The number of vehicles on a road can easily be in the order of thousands, while the size of MANETs is usually smaller.

- Variable Network Density: In a VANET, network density is influenced by the transportation patterns of a society, as evidenced by the difference in traffic in different

areas at different hours during the day. Therefore, algorithms that excel in sparse networks would be disastrous during peak hours, and vice versa. Algorithms deployed in VANETs then need to be able to work properly in ever-changing network densities or degrade gracefully at the very least.

On the other hand, network density is relatively stable in a MANET, which allows the network designer to select an algorithm that will take advantage of the specific nuances of the MANET at hand (Jones et al., 2001; Tseng et al., 2002; Mizumoto et al., 2004).

- Computational power: Related to the issue of energy, since there is a correlation between computational power and energy cost (de Meulenaer et al., 2008), a node in a MANET has limited computational abilities. For this reason, some MANET protocols do not encrypt messages (Groat et al., 2011), as it requires a great deal of calculations and consumes energy. On the other hand, a VANET is not restricted to energy, but its computational power is restricted by the ephemeral nature of the network. Since the time period of a link between nodes in a VANET is very short, any manipulation of a message that takes too long could render transmission attempts futile.

Due to these differences, the protocols used currently in MANETs are unsuitable for VANETs, and research done in the field of MANETs is not easily adaptable or transferable to VANETs. Vehicular ad hoc networks have considerable challenges that require new approaches and protocols.

### 1.1.3 Applications

The ultimate purpose of the deployment of VANETs is to improve the driving experience and safety of drivers and passengers. By being better informed of their environment, such

as road conditions or changes, drivers can react earlier or preemptively to changes on the road. Moreover, by allowing vehicles to communicate, it is possible to send assistance to those that require it in a more efficient or prompt manner. There are many applications envisioned for VANETs, which are mainly divided into two categories: *safety-enhancing applications* and *commodity applications*.

**Safety-enhancing applications** are designed to decrease the probability – and by extension, the number – of driving accidents. Since a vehicle can move at speeds that far surpass human motion, drivers may not have sufficient time to react to sudden changes in traffic or road conditions. Safety applications should give warning to drivers in advance, so that they can change lanes, stop, or warn others, thus avoiding accidents or limiting their damage.

Moreover, communications could be designed to shorten the response time of emergency vehicles. For example, in an accident where the driver loses consciousness, the vehicle can broadcast a warning message that the driver is allergic to certain medications or is diabetic, so that when the ambulance arrives, the paramedics have a better idea of how to deal with the situation and provide more adequate assistance. In less dramatic scenarios, if a car has to stop due to a flat tire on the highway, it could ask for the response vehicle to come with a tire-patch kit, instead of having to bring a tow truck to pick up the vehicle. This could prevent road congestion, saving people time and decreasing fuel consumption.

In safety enhancing applications, security is mandatory, since unauthorized modifications of the message by vehicles or RSUs could result in catastrophic outcomes. Messages that contain information that is of public interest (e.g. road conditions, traffic rate, etc) do not need to be encrypted, since the data should be readable by all without unnecessary procedures that slow down the processing of the data.

On the other hand, with messages that contain a passenger's private information (e.g. medical records, driver's license, etc.) it is imperative to encrypt that private information. Otherwise, every node that receives a copy of the message through the routing process could read the information. Knowing this, users would be reluctant to use the system.

**Commodity applications** Commodity applications provide drivers with discretional information and services, such as the location of and prices at gas stations, remote payment of toll booths, on-demand entertainment and advertisement.

An envisioned use of this system would be the case of a driver on a highway who is running low on fuel. The system inside the car notices this and locates the nearest gas stations. The system can filter and sort the locations based on parameters set by the driver, such as fuel price, brand, distance from the highway, and it can also provide up-to-date offers and deals offered by the gas station.

While they can provide more comfort to the passengers inside a vehicle as well as useful information, commodity applications do not enhance the safety of the vehicle or the passengers inside it. For this reason, algorithms that assign priority to messages have been developed (Xu et al., 2004) so that security-enhancing application data is not overtaken by information from commodity applications.

Security in commodity applications is on an as-needed basis. While messages that send private information – such as credit-card information for making payments – need to be secured, informational messages – such as gas prices – do not need to be encrypted to be useful.

## 1.2 Routing in VANETs

At the very core of ad hoc networks such as VANETs and MANETs lies the routing protocol. This is the algorithm that achieves communication among nodes in the network, and that tries to do so in the most efficient manner, while following the assumptions and restrictions of the network. For VANETs, there is a category of routing algorithms that broadcast messages over an area defined by the sender of the message. This class of geographical broadcasting (*geocasting*) algorithms is appealing because broadcasting increases the probability that the message will arrive to its destination, while simultaneously decreasing the message overhead associated with broadcasting by virtue of restraining the geographical scope of the message to the area of interest. While there are several existing geocasting algorithms for VANETs, we find that they are designed to ensure either privacy or reliability, but not both.

## 1.3 Motivation

There are several routing algorithms that can be used for communications in a VANET (Casteigts et al., 2011; Schoch et al., 2008; Li and Wang, 2007) which have different approaches to achieve different goals. However, we find that they are designed to ensure either privacy or reliability, but not both.

Existing geocasting algorithms that focus on message privacy (El Defrawy and Tsudik, 2008; Festag et al., 2010) cannot ensure that the message will actually arrive at its destination, making them private but not reliable. Conversely, algorithms that ensure that a message arrives at its proposed destination (Basagni et al., 1998; Stojmenovic et al., 2006; Schoch et al., 2010) do not take privacy into account, making them reliable but not private.

We argue that both privacy and reliability are required attributes to have useful communications in a VANET. Private messages that never arrive only serve to waste network resources during transmission, while insecure messaging limits the network applications to generating and providing general information only.

## 1.4 Security

For a VANET to be secure, we strongly believe that it needs to have Privacy, Confidentiality and Reliability.

### 1.4.1 Privacy

Encrypting data ensures that unauthorized entities are unable to decipher the message. While encryption is used to transfer private data securely across a network, encryption alone is not able to protect the privacy of the users at both ends of the conversation (Raya and Hubaux, 2005). In the case of VANETs, the users are passengers inside the vehicle, who form a large part of the general population and have varying levels of technical knowledge and/or comfort with releasing private information.

If the communication patterns of a specific vehicle can be tracked and analyzed, a great deal of information can be surmised about the driver behind the wheel. Information such as driving habits – how fast or slow the vehicle is driven, daily route to work – can then be used to extract information about the driver without his or her knowledge or approval.

While it may be argued that most of this data can be collected by other means (e.g. following the car closely, accessing government record, etc.) the main concern with this technology is the ease that it provides for data collection and analysis. The

same argument can be made in other fields where the transfer of data is done using a method that addresses privacy concerns (Langheinrich, 2002). Regardless of the field of application, newer technologies that are pervasive in the user's lifestyle need to take the privacy of the user seriously. Failure to do so can lead to catastrophic loss of data (White et al., 2011).

There are situations in which it is acceptable to release private information to certain entities. For example, a law-enforcement agency may be interested in tracking a delinquent or stolen vehicle to bring a criminal to justice, or an academic research group may be interested in the daily traffic of a neighbourhood to develop a better road design. In scenarios such as these, there is an implicit or explicit acknowledgement or agreement by the users to release the information to the interested parties, which are also bound by strict protocols to maintain the confidentiality of the data.

Nevertheless, while users may be willing to release the information to such entities, they would be unwilling to extend the same trust to individuals with malicious intentions or groups that could unfairly profit from this information. For example, by reading the location where a driver parks its car, an enterprising person could model the shopping patterns of an individual (e.g. buys groceries every Thursday, goes to the movies once a month, etc.). If a business is able to have such information readily available, it could very easily tailor its advertising – or even pricing – to that individual to make a better profit. Say a car-insurance agency is able to obtain the driving patterns of and information on an individual it insures. Based on an analysis of the driving habits – such as speeding patterns, areas through which the vehicle is driven, or even where the vehicle is usually serviced – it could decide to increase the insurance premiums, or even refuse service altogether. Most drivers would be justifiably unwilling to share this vast amount of private information with such a group.

These privacy concerns could lead to legislation being created and enforced – with varying degrees of success –, but if the technology can be designed to be intrinsically and automatically private, it removes the need to have human intervention to correct the issue manually. Therefore, the concept of privacy encompasses more than just encrypting data. We then define privacy as the ability to transmit data in the network without releasing information that can be used to gather data about a vehicle – such as identity, position, speed and trajectory – by a third party.

### 1.4.2 Confidentiality

Confidentiality in VANETs is closely related to privacy: it is the idea that the message information can only be read by the intended destination node, and not by nodes that transport the message. Currently, most VANET applications concentrate on providing information of public interest, such as road conditions or traffic levels, in which the message content can be read by any vehicle that receives the message. This decision, however, is invalid for applications that handle confidential information, and limits the usefulness of the network in general. Future uses of the network, such as paying toll booths remotely, cooperating with a police officer, or even verifying that a vehicle is a police vehicle, are not going to be feasible or acceptable if the information can be read by every vehicle in the network. Therefore, communication algorithms that wish to be versatile need to allow for confidential communications. This is commonly achieved using asymmetric encryption schemes.

However, as already mentioned (Raya and Hubaux, 2005), encrypting the data is not enough to achieve privacy. Unless privacy is strong, drivers would be justifiably reluctant to adopt this technology.

### 1.4.3 Reliability

Even when assuming ideal physical scenarios, due to the ever-changing topology of an ad-hoc network, it is possible that messages may arrive too late to their destination, or not arrive at all, due to temporary disconnections or a large latency. Therefore, algorithms have been modeled to offer a level of certainty (*reliability*) for the traversal of messages from the source to the destination. This philosophy is in sharp contrast to best-effort approaches, where the arrival of a message is never assured.

Reliable protocols usually achieve this by creating multiple paths between the source and the destination, creating paths with as few hops as possible, or a combination of both.

## 1.5 Geocasting

Geocasting (Casteigts et al., 2011) is a form of one-to-many communication. Message destination is defined geographically (i.e. *where* instead of *who*), and only vehicles within a specified region are allowed to read the data.

An example of where it may be useful to have private and reliable geocasting is the case of a driver who notices an accident and informs the nearest hospital. The notification may contain further details about the damaged vehicle – license plate, vehicle make, colour, etc.– depending on the sensing capabilities of the sender. This information would need to be kept confidential, so that only the hospital can benefit from the information received. The hospital does not need to know anything about the notifier itself (location, speed, etc.) to receive the information, and the notifier is only interested in knowing the hospital's coordinates to send the information.

Once the ambulance arrives at the scene and after assessing the situation, it could decide to move the accident victim to the hospital. The ambulance then sends a request to the hospital that contains patient information (age, blood type, condition, etc.) to speed up the check-in process at the entrance. Since this information is confidential, it needs to be encrypted so that only the hospital can decode it.

Another scenario where anonymous geocast communication is very useful is when a vehicle wants to warn the vehicles following it of bad road conditions (e.g. ice, accident, etc.). The sender does not know which specific vehicles might be interested in this information but knows that nearby vehicles within a certain range might find it useful, so it sends a message to that area by a geocast. Conversely, vehicles in that area do not need to know the identity of the benefactor to receive the information.

In direction-based broadcast algorithms (Basagni et al., 1998; Stojmenovic et al., 2006) the geocast region is the area where the destination node is within a certain probability. This region is defined by a direction vector and an angular range, which are used to obtain a diameter and tangent line of the circle encompassing the geocast area. On the other hand, algorithms that afford reliability, such as the DSG protocol (Schoch et al., 2010), may employ a probabilistic forwarding approach to limit message duplication to deter malicious vehicles from overloading the network, especially in saturated networks. In DSG, a vehicle receiving a message decides to forward it with a certain probability, which is heavily influenced by the behaviour of the message sender. However, to disseminate a message, those algorithms use topology knowledge, such as the mobility of the destination vehicles or the neighbouring nodes of said destination.

## 1.6 Our Contribution

In this thesis, we are going to achieve both requirements (reliability and privacy) using geocast-based communications.

We propose two new private and reliable geocasting algorithms. The first uses a probabilistic model with adaptive traffic restriction and dynamic probabilistic forwarding to obtain reliability, a combination that the DSG protocol (Schoch et al., 2010) also employs. To obtain privacy, the traffic limitation and forwarding algorithms use parameters that do not uniquely identify a node, to generate pseudonymous and unlinkable (Pfitzmann and Köhntopp, 2001) communication channels. Unlinkable channels are those in which it is impossible for a network observer to know the ID of the source and/or destination of a message. When a node receives a message, instead of storing the identity of a sender node when forwarding, it calculates the direction vector of the message, and make the forwarding decision based on the message traffic generated in the message's source area, as well as the angle of incidence.

The dissemination method is a variation of direction-based geocasting, similar in appearance to a protocol designed by Basagni et al. (1998). However, while Basagni et al. derive the angle parameter from a fixed probability, to obtain a tangent of the destination area, our work uses a preset angle value, which controls the probability of forwarding a message by receivers, and by extension manages the messaging overhead.

The second is a deterministic protocol that uses a transmission-area coverage algorithm technique used in wireless sensor networks for energy conservation and to avoid message duplication (Simplot-Ryl et al., 2005). The calculation of the delay varies depending on the forwarding task at hand, whether it is geocasting or routing. To achieve unlinkability, the forwarding nodes only release the current location of the node, with no other personally identifiable data. Upon receipt of a message, the forwarding node

decides the forwarding delay using only the position of the sending node in combination with the destination area.

While the second algorithm uses the same headers as the first, the angle parameter $\theta$ is used instead to define the broadcast coverage area of interest. Moreover, in both algorithms, messages are encrypted asymmetrically to have message confidentiality and authentication, as well as to allow future applications that require access control. This is achieved using Public Key Encryption (PKE).

### 1.6.1 Assumptions

The routing algorithms proposed in this thesis work best when sending messages in a straight line. While they are able to turn corners, their ability to route a message is diminished, and reliability decreases.

For the adversarial model, we assume that an attacker will try to gather as much data from a particular node of interest as it can by way of trailing the node, or setting itself up in a pre-selected position in the network. Moreover, attackers could try to disrupt routing attempts by refusing to forward messages, or failing to do so in a timely fashion (blackhole attacks). We assume that by using broadcasting techniques, it is possible to deter blackhole attacks (Ramaswamy et al., 2003; Buttyán and Hubaux, 2007).

Attackers could also try to flood the system with replay attacks, by constantly repeating an outdated or malformed message. Finally, attackers could try to mount Sybil attacks (Douceur, 2002), in which they pretend to be a different node in the network. There are two types of attacks that an adversary could try in this scenario; passive attacks that collect private information, and passive or active attacks that disrupt communication. We prove that our algorithm is effective in defending against this form of attack.

Table 1.1: Notation used in the paper.

| Notation | Meaning |
|---|---|
| $N$ | Number of nodes in the network |
| $n_i$ | $i^{th}$ node |
| $R_i$ | $i^{th}$ RSU |
| $C_i$ | $i^{th}$ CA |
| $A_i$ | $i^{th}$ geographical region |
| $l_i(t)$ | Location of $n_i$ at time $t$ |
| $Q$ | Queuing groups |
| $q_i(n)$ | $i^{th}$ group of $n^{th}$ node, $q_i(n) \in Q$ |
| $M$ | Set of all messages |
| $m_i$ | $i^{th}$ message $m_i \in M$ |

### 1.6.2 Notations

For the purpose of this thesis, when describing the algorithms the terms *node* and *vehicle* can be used interchangeably. Whenever communications are meant to extend to an RSU, or other type of static node, it will be explicitly designated as such.

When describing a message, the message *content* is referred to as the *data* being transmitted, to differentiate it from the full message packet, which also contains transmission headers and/or extra bits necessary for the routing algorithm. Table 1.1 gives the notations that we follow throughout the rest of the paper.

### 1.6.3 Adversary

We consider the following type of attackers:

- **Single attacker with local knowledge of the network:** An attacker can only directly communicate with the nearest neighbour. No single attacker has global knowledge of network topology. Moreover, multiple attackers are non-colluding.

  With this assumption, we remove the possibility of mounting a wormhole attack.

- **Insider attack:** An adversary can authenticate itself as part of the network, and therefore has valid credentials for communicating.

  In the simple scenario of an outsider attack, the adversary does not have valid credentials for communication. Therefore, any messages it generates are immediately discarded by a node in the network.

- **Attacker knowledge:** Since it is an insider adversary, it is well versed in the protocols and algorithms used for communicating in the network. For this reason, if there is a known weakness in the encryption algorithm, it can be assumed that the adversary is aware of it, and can exploit it effectively.

- **Active Attacker:** The adversary can listen to the communications, and can read the message headers, modify them, and selectively inject them to the network, to interfere with communications.

  Adversaries will try to gather as much information as possible about a specific vehicle by storing the data packets that the vehicle sends. The passive attacker will only be able to read the message headers, and store them for forensic analysis.

- **Finite amounts of calculation power and storage space:** An adversary does not have access to unlimited processing power that could, theoretically, allow such

an adversary to decrypt the messages in finite time. Moreover, the adversary can only store a finite – though possibly large – number of messages.

- **Mobile attacker:** An adversary can be mobile, and it could potentially move on a random walk. However, the most efficient – and common – attacker would be one that moves along the pre-established paths in the model (i.e. driving on the roads and streets, following proper driving behaviour), since those attackers can follow a targeted vehicle closely, while being inconspicuous.

  The random walk mobility model can be easily ascribed to an individual with a transmission device that can move about the network by foot or other conventional means of transportation (e.g. a helicopter or all-terrain vehicle). From a network perspective, the behaviour of these attackers can be easily distinguished from the behaviour of the vehicles that follow the normal pathways.

  On the other hand, an adversary could be fixed in place. This scenario is more likely when the adversary is trying to create counterfeit RSUs, such as traffic lights or informational stations.

## 1.7 Organization

This thesis is organized as follows:

- Chapter 2 is a background on vehicular ad hoc network technology, as well as security research applied to the field. This chapter presents the research previously done in the fields related to this thesis. We present work that has been done in direction-based geocasting, broadcast traffic control and transmission-area coverage. We also present a brief description of current security protocols for VANETs.

- Chapter 3 presents a geocasting protocol that uses a probabilistic approach to direction-based geocasting to achieve privacy and reliability. The algorithm also prevents flood and blackhole attacks, as well as deters Sybil attacks on the network. The simulation results provided serve to illustrate the effectiveness of the protocol in VANETs.

- In Chapter 4, we introduce a different geocasting protocol that uses a deterministic method combined with a transmission-area coverage algorithm, which also achieves privacy and reliability. This algorithm reduces the overhead caused by duplicated messages, as well as provides a higher level of reliability than the previous algorithm. Once again, the simulation results are provided to demonstrate its effectiveness in VANETs.

- Chapter 5 discusses the encryption scheme used in both algorithms. It details the methods used to manage, release and accept certificates and encryption keys. In this chapter, we also provide proofs that our algorithms achieve their privacy goals.

- Chapter 6 contains a conclusion and discusses possible ways in which the work in this paper can be improved and expanded upon.

# Chapter 2

# Related Work

We now discuss the research done in the fields of geocasting and security in VANETs. Where appropriate, we also note where research in the two fields intersect.

## 2.1 Efficient Geocasting and Traffic Restriction

One approach to obtain reliability is to generate many communication routes, or channels, between the sender and the receiver. However, the routes should minimize the number of nodes they have in common, to avoid sending duplicate messages or running the risk of being affected by blackhole attacks. To solve this, gossiping protocols forward a message using a probabilistic (as opposed to a deterministic) algorithm. This way, the message travels in a pseudo-random path through the network. Moreover, a probabilistic forwarding algorithm improves its efficiency if the value of the forwarding probability $p_{forward}$ is dynamically calculated and updated to adapt itself to the network requirements.

These concepts are applied by Bako et al. (2008) in the Advanced Adaptive Gossiping protocol (AAG), which calculates and updates (adapts) the forwarding probability $p$

every time a message is received. Each receiving node $R$ in the network stores and refreshes its two-hop neighbour information. When a message is received, $R$ calculates the set $M_r$ of neighbouring nodes that have received the message directly from source node $S$. The set $C$ of children nodes would then be the two-hop neighbours of $R$ that did not receive the message from $S$, and the nodes in $M_r$ that are connected to $C$ would be the set of parent nodes $K$. In short,

$$\begin{aligned} M_r =& \{m \in neighbor(R) | m \in neighbor(S)\} \\ C =& \{c \in neighbor(R) | c \notin M_r\} \\ K =& \{k \in neighbor(C) | k \in M_r\} \end{aligned}$$

With the size of $K$ known, $p_{forward}$ is then calculated to be

$$(1 - p_{forward})^{|K|} < (1 - \tau_{rel})$$

where $\tau_{rel}$ is proportional to the network diameter $\sigma$ and a predefined minimum reception rate $\tau_{arp}$, so that $\tau_{rel}^{\sigma} = \tau_{arp}$

The elegance of AAG lies in the fact that it not only has a high level of reliability, but it also adapts well to dynamic network topologies and densities, such as VANETs. Nodes in high-node density areas will reduce their message forwarding rates, while forwarding more in less dense areas.

The DSG protocol by Schoch et al. (2010) was designed with reliability as its goal, and makes use of AAG to obtain this. It is a geocasting protocol in which messages are signed to ensure that they are not modified, and they are disseminated using AAG. To further reduce message duplication, and to avoid flooding attacks that could cause congestion, the protocol applies an adaptive load-control algorithm. When a node $R$

receives a message from a source node $S$, it calculates the historic traffic use that $S$ has imposed on $R$. It does this by determining certain load factors $k_i$, such as the messaging frequency of $S$, the cumulative payload of messages coming from $S$ and the cumulative message payload of the destination area. Moreover, Schoch notes that different values and measurements could be used to improve detection of network bandwidth abuse.

Once these values are calculated, they are assigned weights $w_i$ to normalize them, and the resulting product is used to determine the network traffic $L$ created by node $S$. The value of $L$ is then

$$L = \prod_i w_i k_i$$

With this value, $R$ then calculates the fluctuation of $L$ caused by $S$ over a period of time. Node $R$ sets minimum and maximum network traffic thresholds $[\theta_{min}, \theta_{max}]$ and subdivides it into $j$ intervals or steps. Each step is of size $(\theta_{max} - \theta_{min})/j$, and it is assigned a specific forwarding probability $p_j$, where

$$p_i < p_k,\ 0 < i < k < j$$

In other words, steps closer to $\theta_{min}$ have a higher value for $p_i$, and those closer to the other end have a smaller value. The load $L$ is then placed inside one of the steps, and the associated forwarding probability $p_i$ is given to the message received from $S$. Using this method, node $R$ can refuse to forward messages from $S$ with a probability proportional to the message traffic it creates. Demanding nodes have their traffic limited to reduce their burden on the system, while less demanding nodes have most of their messages forwarded successfully. This way, malicious nodes that try to flood the network by sending duplicates fail to saturate the network, and the network is protected against flooding attacks. By combining traffic control and gossiping, DSG is very reliable, efficient and protects itself against blackhole and DoS attacks.

However, DSG has privacy drawbacks. The protocol tracks the uniquely identifying node information (location, total message payload, etc.), while AAG relies on storing and refreshing the network information (two-hop neighbour information) to efficiently avoid repeating a message. These decisions create a privacy issue, in which a malicious node could deduce the trajectory of a given node by keeping track of the messages it is sending, the rate at which it is sending them, and the location when the message is being sent. Furthermore, while the authors propose to sign the message for authentication purposes, they assume that the messages are not delivering confidential information, and thus do not encourage encrypting the data itself.

This severely limits the uses of the network, since nodes that disseminate messages with confidential data can read the information even if they are not the intended destination, which might not be desirable in certain cases. For example, a car in an accident could send a report to the nearest hospital with the driver's age and medical record. While this is critical information to save the driver, everyone forwarding the message should not be able to read this information.

## 2.2 Direction-Based Geocasting and Position-Based Routing

The idea of direction-based geocasting algorithms is to send a message along several routes in a direction that leads towards a geographic area. For example, the DREAM algorithm by Basagni et al. (1998) is a directional algorithm that is a hybrid between proactive and reactive protocols in a mobile ad-hoc network. When node $S$ sends a message to destination $D$, it calculates an angular range based on the mobility information it has received on $D$. The range is calculated to provide a radius of the maximal position

of $D$ within a certain probability. The messages are then forwarded to all neighbours in the direction of the range.

We use the concept of setting an angle of spreading, and also see the positive effect it has with the probability of reaching an area. However, we use the value to reach a geographic area instead of a specific node, and its effect on the probability of success is different. In the work of Basagni et al. the probability $P$ of *reaching a node* is fixed, and it drives the value of $\alpha$, which represents the angular range of mobility of a node of interest. In our work, the value of $\alpha$ is fixed, and it affects the probability of *forwarding a message*, which as a byproduct affects the probability of reaching an area.

The privacy issue in DREAM is that every node maintains a table with the mobility information of the nodes in the network. This information can be easily organized to provide information about the vehicle's driver, like their driving patterns or even their home address.

On the other hand, the VD-GREEDY and CH-MFR algorithms by Stojmenovic et al. (2006) are routing algorithms optimized for mobile ad-hoc networks (MANETs). The main idea in the routing process is to optimally forward the message to the neighbour that is closest to the destination (i.e. progresses towards the destination). In VD-GREEDY, these neighbours are determined using the Voronoi diagram of neighbours that are near the destination area, while CH-MFR uses the convex hull of neighbouring nodes to select the node to forward the message to.

The algorithms presented by Stojmenovic et al. have a smaller flooding rate compared to directional algorithms like DREAM. Moreover, the selection of forwarding nodes is optimal with respect to power consumption, which is a concern in MANETs. As stated earlier, power consumption is not a concern in VANETs, in which nodes are connected to a generous power supply (e.g. the engine of the vehicle). That being said, reducing the

number of duplicated messages also decreases the load on the system, which is desirable in VANETs as well.

The main privacy concern with these algorithms is that they require a high level of topology knowledge – the neighbouring node's positions – to effectively select a node to forward to. Once again, this information could be used by a malicious node to make deductions about a specific node's identity.

## 2.3 Transmission Area Coverage

In area-coverage problems, a set of nodes is placed in a given area, in which each node is able to cover a circle with a radius centered at the node itself. To efficiently communicate, we need to select the minimal set of nodes that covers the entire area. The resulting set is then the Connected Dominating Set (CDS), or the set of nodes that need to broadcast the message for all nodes in the area to receive it in one hop. This model has been studied in sensor networks (Simplot-Ryl et al., 2005; Gallais et al., 2008; Gobriel et al., 2006; Ye et al., 2003) to provide redundancy while minimizing energy expenditure. Moreover, this research is also applicable in MANET routing, usually in the form of connected dominating sets (CDS), to provide an approximation of minimal connected dominating sets.

Simplot-Ryl et al. (2005) surveyed several algorithms in which sensor nodes can independently decide to turn off their transceivers (become inactive) when their communication area is effectively covered by neighbouring nodes. Tian and Georganas (2004), propose sensor-area covering schemes in which each node decides whether or not to remain active with a fixed probability, the value of which is derived based on the expected percentage of the sensing-area coverage and depends on neighbours announcing their position and communication range.

Ye et al. (2003) present the PEAS algorithm, which is a localized protocol that works on asynchronous sensor networks. They assume that all sensors have the same sensing and transmission radius $R$, and that the minimum distance between nodes is the sensing radius. Initially, all nodes are inactive for a random period of time. After the sleeping period is over, nodes send a probe to see the neighbours that cover it within a certain range $R_p$, and active nodes that receive the probe send back a reply. While the node does not receive a reply from their probes, they remain active and become inactive for another random period of time upon receipt of a reply. The sleep time is calculated using an exponential distribution function that is constantly adjusted depending on the previous activity and inactivity periods of the node at hand. The authors note that smaller values of $R_p$ will result in a higher number of active nodes and thus provide a more redundant and robust network. Furthermore, to avoid disconnections, it is suggested that the value of $R_p$ be smaller than the transmission range.

Gallais et al. (2008) extend this concept and apply it on a routing algorithm based on a timeout scheme. Periodically, every node selects a timeout to listen for communications. Before the time expires, nodes receive and store the position of and decision made by nearby nodes with shorter timeouts. Once the timer is finished, nodes calculate the effective coverage area by the neighbouring nodes. On subsequent communications, nodes can quickly decide to not forward a message when they know that it is fully covered by other nodes.

While VANETs do not benefit from the energy savings provided by these models, they do profit from the connectivity that is offered by the coverage, as well as the reduction of duplicate messages being created.

## 2.4 Security and Privacy in VANETs

There is extensive literature regarding security (Raya and Hubaux, 2007, 2005; Anantvalee and Wu, 2007; Stajano et al., 2007) that details what requirements should be in a VANET for it to be considered secure. Papadimitratos et al. (2008) have listed some of these, which include message authentication, integrity and confidentiality. In most routing models, while the first two can be achieved by signing the document, the last requirement implicitly demands the use of encryption, whether by generating shared secret keys (Papadimitratos and Haas, 2002), using symmetric or asymmetric keys, and attempts to induce privacy by using pseudonyms. DSG is found to be insufficient in this part, since messages are assumed to be of general interest and are never encrypted.

A protocol that does encrypt data and ensure privacy is PRISM by El Defrawy and Tsudik (2008), which encrypts data using group keys and sends messages to a geographical area rather than a specific node. The routing itself is done using a modified version of AODV, where the sender adds headers to the RREQ containing the DST-AREA and a timestamp. The data is encrypted using a one-time use public key, and the message is signed using a group key. Upon arrival to the destination, the one-time public key is used to create a shared secret key, and a secure communication channel is established between the two parties. To enhance privacy, PRISM limits the number of nodes that take part in the routing process and establishes as few routing paths as possible.

However, while an adversary in the routing path cannot glean much information about the sender or the receiver, there is nothing that prevents it from doing a black-hole attack and reducing message reliability. By refusing to forward the message or by delaying the forwarding of the message, it is effectively disabling the route that is being established or sending a message when it is no longer useful. Another weakness of the protocol is that the DST-AREA header itself is not encrypted, so that it can be read by

everyone, and could easily be modified. Moreover, there is no timely way to deal with adversarial agents. The RREQs are stored, and would conceivably be used for detecting whether a node has been abusing the system, and alert the network of this adversary. However, this is done *ex post facto*, and does not protect the network when the attack first happens.

Priority-Based Routing (PBR) is another geocasting algorithm by Harsch et al. (2007). At its core, PBR is composed of three elements: beaconing, node-location tables and forwarding. Whenever a node is interested in a specific node, it will store its approximate location in a table. Nodes of interest are destinations, routing nodes and neighbours, who will periodically send their locations by beacons.

To achieve this, when a node realizes it is of interest to another node, it will offer its approximate location. The reason why the location is only an approximation is because it is acknowledged that the vehicles could be moving at varying velocities (speed and/or direction), so the position offered is where the vehicle is most likely to be by the time another data packet is sent by the source.

The data itself is encoded end-to-end, so that only the destination node is able to read it, and hop-by-hop, so that only the node the message is forwarded to is able to handle it. To enhance privacy, the routing process separates the data from the routing headers and encodes it separately. The data is only readable by the destination (end-to-end) and is thus immutable, while the routing information is read and modified by the routing-path nodes (hop-by-hop) and is thus mutable.

To accomplish this, the algorithm uses two steps of encryption: end-to-end data is encrypted and signed by the source nodes, while hop-by-hop data is encrypted and signed by the sender nodes. On receipt of a packet, a forwarding node updates the mutable field values and re-forwards the packet. The destination node verifies both the sender

and source signatures before accepting the data. To reply to a message, the same process is used, making the system work like two half-duplex communications. For all practical purposes, no specific route is ever established or maintained.

This algorithm provides a solution for PRISM's DST-AREA problem, in that if any sender were to corrupt that header, as long as there are non-adversarial nodes forming a path to the destination, the message will arrive, protecting the network from wormhole attacks. Moreover, since it uses hop-by-hop encryption, it is easier to detect rogue nodes. However, caching the location and the certificates to make the algorithm more efficient still results in a loss of privacy. Since this algorithm keeps track of the position of the communicating nodes, it is feasible to determine – and predict – the movement patterns of any given vehicle.

The Secure Routing Protocol (SRP) by Papadimitratos and Haas (2002) is designed to be applied over existing reactive routing protocols, such as AODV and DSR. A strong assumption SRP makes is that the source and destination have already established a way to recognize each other beforehand, which the authors call a security association. For example, the two communicating parties can generate a shared secret key using the public keys of each side.

In the SRP algorithm, a source node $S$ sends a RREQ message to the destination node $T$. This RREQ message has two extra headers for identification purposes: a query sequence number and a random query ID, used to detect duplication and reordering. The message authentication code is calculated using the source, destination, RREQ ID and the shared key as parameters. When $T$ receives the RREQ, it tries to generate as many paths as possible in order to avoid disconnections, which could be caused by malicious nodes or simply by nodes leaving the network. To deter malicious nodes, SRP implements a ranking system in which nodes that send the highest number of queries

are given the lowest rank. Nodes prefer to talk to neighbouring nodes that have a high rank, and queries received from low-ranked nodes are given less priority.

This algorithm deals with non-colluding malicious nodes by isolating them from the routing process. Also, since secret keys are used, privacy is ensured by avoiding IP spoofing. However, any routing protocol that uses caches – such as DSR – will suffer from cache poisoning from the malicious nodes (Michiardi and Molva, 2004), making caches useless and reducing the efficiency of the protocol. Moreover, two malicious nodes working together can distort the perceived network topology by mounting a wormhole attack. While having adversarial nodes colluding is another problem in itself, since this algorithm assumes that there is a pre-established security association, it invites nodes to acknowledge each other before initiating communications, which simplifies the task of creating collusion.

A secure VANET should not only limit itself to encrypting the messages so that only the intended recipient can decode them (message confidentiality), but should also take into consideration privacy issues (Papadimitratos et al., 2008; Freudiger et al., 2007; Raya and Hubaux, 2005). In this sense, privacy protection refers to minimizing the information that can be inferred about a node from its behaviour. The problem with privacy is that, while the data itself is safe from prying eyes, if messages can be linked to a particular vehicle, there is some information about the vehicle – and in turn, the driver – that can be inferred, such as driving habits (speeding or sudden braking) and places of interest (which gas station does this driver frequent), just to name a few options.

To correct this, Papadimitratos et al. (2008) suggest using pseudonyms, which obfuscates the identity of the vehicles when sending messages, while still allowing operations that require identification for security purposes, such as signing messages. The pseudonyms are obtained periodically through a certificate authority (CA), which is the

only entity capable of establishing a link between the pseudonyms and the unique identity of the vehicle, which is kept secret during regular communications. To provide privacy, pseudonyms are used only during a certain period of time, and then promptly discarded, to avoid scenarios in which an adversary could place a vehicle in different locations at different times and observe a connection or pattern.

## 2.5 Encryption and Authentication

Regardless of whether communications are private or not, when a message arrives to its destination it needs to be authenticated. Authentication in VANETs is the process of verifying that the node that created the message is the same as the one declared by the message. Through authentication, the destination can confirm that the message was created by the alleged sender of the message and was not modified or tampered by anyone else.

If a user were to use only a unique identity (real or otherwise), it would be trivial to track its movement and private information. To have secure communications, it is ideal for a node to be able to communicate without being identifiable, which is the basis for anonymity. However, complete anonymity makes it very difficult to authenticate the creator of a message, and as described by Papadimitratos et al. (2008), authentication is also an important feature in secure communications. To allow for both privacy and authentication, a combination of pseudonyms and group signature schemes can be employed.

### 2.5.1 Pseudonyms

Pseudonyms (Pfitzmann and Köhntopp, 2001) are aliases that nodes use to communicate while retaining their real identity hidden, which allow them to retain their privacy. The unique identity is known only to the user and the certification authority (CA) that issues the pseudonyms. When a node communicates with another node on the network, it shows only the pseudonym, and the receiver of the messages can only identify the source by the given pseudonym. When generated appropriately, the real identity of the user cannot be extracted from the pseudonyms. The opposite is not necessarily true, which is all the more reason to protect the unique identification of the node. The use of pseudonyms in VANETs has been extensively studied in research articles such as (Buttyán et al., 2007; Calandriello et al., 2007; Freudiger et al., 2010; Raya and Hubaux, 2005, 2007; Sampigethaya et al., 2007).

To effectively preserve location privacy, a node uses several one-time-use pseudonyms which are discarded once a period of time has elapsed. Frequent change of pseudonyms ensures higher privacy, but pseudonyms are expensive to generate, since they are often obtained only from the CA. For this reason, it may be desired to use the pseudonym for the longest period of time that it is effective at providing privacy. Freudiger et al. (2010) performed a detailed study on the age of pseudonyms and discussed different parameters, such as cost of pseudonym replacement, network density and rate of meetings, to determine when a pseudonym should expire.

The decision to discard a pseudonym needs to be done in a way that does not allow an observer to make a connection between the previous and the new pseudonyms, to avoid being traced. To do this properly, Freudiger et al. (2007) point out that the pseudonyms should be changed only in *mix zones*. Mix zones are areas where nodes cannot be observed or easily differentiated, either by another node or by a RSU.

Beresford and Stajano (2004) show that the level of privacy offered by mixed zones is inversely proportional to the number of vehicles in the area. If there is more than one vehicle in the mix zone and they change their pseudonyms, it is difficult to determine which pseudonym corresponds to which vehicle. However, if there is only one node in a mix zone when it changes its pseudonym, it is clear that the two pseudonyms belong to the same node. Buttyán et al. (2007) show by simulation how the privacy level changes with node density.

Sampigethaya et al. (2005, 2007) suggest another method for deciding when to discard the pseudonym: by using random silent periods between the changes of pseudonyms. This method is effective when working under the assumption that the vehicles are moving in a group with similar speeds. When a vehicle joins the network, it waits a random amount of time before changing its pseudonym. This way, if two nodes enter the network at the same time, they will change their pseudonyms after a random time interval, making it hard to establish a link between the new and old pseudonyms of a given vehicle.

### 2.5.2 Group Signatures

In most encryption schemes, each user can construct (and verify) the public keys of all the other users in the network. This way, when a node receives a signed message, it can simultaneously verify the signature and check the authenticity of the message.

Following the notion of obfuscating the unique identity of a vehicle, if a group of vehicles were to use a similar certificate to sign their messages, it would be difficult to identify the specific vehicle that generated the message. This is the basis behind the concept of Group Signature Schemes. They were introduced by Chaum and van Heyst (1991) to provide anonymity to the signers, and Boneh et al. (2004a) further suggested the use of group signatures in vehicular networks.

Group signatures (Sampigethaya et al., 2007; Calandriello et al., 2007; Studer et al., 2009) can be applied to sign messages in VANETs, so that the receiver of the message can check that the message has not been tampered with, while still being unable to identify the original sender. Besides the evident usefulness of this property, these schemes also provide authentication, conditional anonymity and non-repudiation.

Nevertheless, the ephemeral nature of the group membership forces the certificates to have a short life-span, which results in a high amount of certificate revocation, a notoriously costly operation (Adams and Lloyd, 2002). Since groups can change very frequently in a city network, this scheme is impractical in such scenarios.

## 2.6 Mathematical Techniques and Signature Scheme Used

We will use bilinear pairings on elliptic curves for our encryption and authentication schemes. Let $G$ be a cyclic group of prime order $q$ generated by $g$. Let $G_T$ be a group of order $q$. We can define the map $e : G \times G \rightarrow G_T$. The map satisfies the following properties:

1. $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G$ and $a, b \in \mathbb{Z}_q$, $\mathbb{Z}_q = \{0, 1, 2, \ldots, q-1\}$.

2. Non-degenerate: $e(g, g) \neq 1$. The bilinear mapping of $e$ is efficiently computable.

We use BLS signatures (Boneh et al., 2004b) on the ciphertext to authenticate the validity of the message.

Authentication proceeds in three steps:

1. Setup: A group $G'$ of order $p$ is chosen by key distribution center (KDC) and each user chooses a secret key $x \in \mathbb{Z}_p$. Let $f$ be a generator of $G'$. $H(.)$ is a hash function, such as SHA-1, which is usually employed for this. The public key of the user is then $X = f^x$, where $x$ is the secret key chosen by the user.

2. Signature: A node wishing to send message $m_i$ creates a signature

$$\sigma = H(m_i)^x \in G'. \tag{2.1}$$

3. Verify: Signature $\sigma$ is verified by checking

$$e(H(m_i), X) \stackrel{?}{=} e(\sigma, f) \tag{2.2}$$

We note that

$$\begin{aligned} e(H(C), X) &= e(H(C), f^x) \\ &= e(H(C)^x, f) \\ &= e(\sigma, f) \end{aligned}$$

This result will be used later in our security algorithm to provide encryption services that are appropriate for our purposes and needs.

# Chapter 3

# Direction-Based Algorithm

In this chapter we present our probabilistic geocasting algorithm. We will describe our geocasting protocol and provide empirical data to support our decisions regarding routing.

## 3.1 The Protocol

When a vehicle sends a message, it defines the $DST\_AREA$ and a point $C$ inside it, as well as the angle of spreading $\theta$, to generate a cone where the base is $DST\_AREA$ and the aperture angle is $2\theta$. Vehicles outside this cone will not forward the message, while those inside the cone have an angle $\delta$ relative to the source position $S$ and point $C$. A vehicle inside the cone then forwards the message with a certain probability, which depends on the $\delta : \theta$ ratio. See Figure 3.3 for a visual example.

Receiving nodes group all incoming traffic into separate groups, depending on the geographical source of the message, and calculate the network traffic that group (or area) is generating. When an area increases its levels of messaging, its traffic is restricted by forwarding nodes.

Each vehicle has many pseudonyms that are given by a trusted authority – a government agency, or an existing certificate authority – when the vehicle is registered. Such pseudonyms are changed from time to time as discussed by Raya (2009). A public/secret key is also given for each pseudonym and stored in the vehicle. These keys are used by RSUs for securely distributing secret credentials for constructing group keys for decrypting messages within a geocast region. Each node has a pseudorandom number generator and a hash function $H : \{0,1\}^* \rightarrow G'$, where $G'$ is a group of order $p$ chosen by the trusted authority. These are used to generate encryption keys to geocast messages. We will discuss the type of secret information as well as the encryption/decryption process in Chapter 5.

Before sending a message, the sender populates the immutable header fields (position, $DST\_AREA, \theta$, etc.), depending on the type of message it wants to send. For example, it might want to send information that a road in area $A_{r+1}$ is to remain closed until 12 pm for maintenance, to enable vehicles in region $A_r$ to change their routes. The sender position field is also filled with the value $l_i(t)$, so that receivers can decide whether to forward or discard the message. For example, if the target area has been passed and the receiver is in front of the sender, the data can be dropped.

After all the headers are generated, the message is encrypted and signed. The encryption and signature schemes are discussed in Chapter 5. The data packet is then broadcast omnidirectionally.

Each message has a message header with the following fields. Figure 3.1 offers a visual reference.

- $DST\_AREA$: The geographic area describing the destination location. It can be any arbitrary description, such as centre and radius for a circular area or width and length for a rectangular area. For any area described, a point $C$ inside the area

| *DST_AREA* | $\theta$ | Timestamp |
|---|---|---|
| Expiration Time | TTL | Sender Position |
| Protected Data | | |

Figure 3.1: Message packet content.
Mutable fields are in light grey, while immutable fields are dark grey.

can be obtained either explicitly from the header (e.g. the centre of the circle) or it can be calculated (e.g. a random position inside a polygon)

- *Angle of spreading* $\theta$: Used to set the propagation ratio of length:width of a message being broadcast in the network.

- *Timestamp, Expiration Time* and *TTL:* Used in tandem to measure the message's validity with respect to time and number of hops. A message that exceeds the validity time window and/or number of hops is discarded.

- *Sender Position:* When a node decides to forward a message, it updates this field with its current location, $l_i(t)$. A receiving node checks this field to determine if it is closer to the destination than the sender.

Upon receipt of a message, a node checks its validity (timestamp, TTL, etc.) to avoid re-sending when unnecessary.

If the message is valid, it checks the network traffic generated by the message's source area. Areas that generate a large number of messages in a short time or that send very large messages are suspected of trying to overload the network by flooding it, so the probability of forwarding a message from that area is reduced. Conversely, messages arriving from areas with moderate traffic are given a higher probability.

Nodes forward messages by broadcasting, which is expensive since it creates many messages. To reduce messaging, the broadcast algorithm checks the position of the sender to see if it is closer to the destination, and it also checks the spread angle $\theta$ to determine

if it can be part of the routing path. If either check fails, the message is discarded; otherwise, it is assigned a forwarding probability.

The algorithm then combines the forwarding probability $P_t(q_i, p)$ calculated by the traffic restriction algorithm with the forwarding probability $P_b(\delta, \theta)$ generated by the broadcast algorithm to constrain the number of redundant routes created by the duplicate messages that circulate in the network.

### 3.1.1 Message Traffic Restriction

A node $n$ would place each incoming message in a group $q_i(n) \in Q$ that is related to the position vector of the message sender with respect to the node. We then define $q_i(n)$ as follows:

$$\text{Let } q_i(n) \subseteq M$$

$$\forall i \neq j, q_i(n), q_j(n) \in Q : q_i(n) \cap q_j(n) = \varnothing$$

$$\bigcup_{i,n} q_i(n) = M$$

In other words, a node can decide to create many discrete directional groups, as long as there is full coverage of the surrounding area, geographical zones do not overlap with each other and a message is found in exactly one group.

More groups could be used to get a more detailed view of network traffic from a specific direction. For example, you can designate four separate directional groups (N, E, S, W) or have twelve groups, each covering a 30 degree arc, or any arbitrary number of groups and coverage for each.

From the perspective of the nodes, each group represents one neighbour (e.g. a *supernode*), and they treat all messages coming from the area associated with that group as such. See Fig. 3.2 for a visual reference.

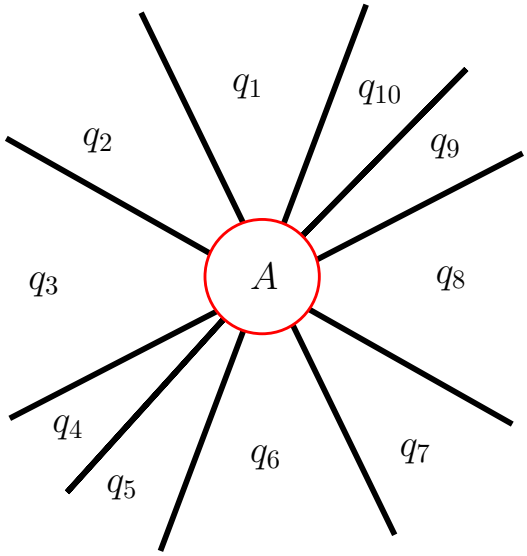


Figure 3.2: Sample design of directional groups.
All messages received are placed in one group, and there is no overlap between areas.
All nodes in a zone are treated as one supernode.

Instead of calculating the frequency of messages and accumulated payload size of each individual sender, the node $n$ would assign weights $w_i$ to network factors $k_i$, such as message frequency and accumulated payload size for each group $q_i(n)$, and calculate the traffic from $q_i(n)$ over a period $p$ using the formula:

$$l_{n,q_i}(p) = \sum_j k_j w_j$$

The value of the weights $w_i$ depends on the desired network configuration. Networks that want to restrict duplicates would place the greatest emphasis on message frequency, while networks that want to be reactive in the short term to data spikes would increase the weight of the cumulative payload.

To avoid a situation in which an adversary can disable all communication from one group or, inversely, flood the system, we define a probability range $[P_{min}, P_{max}]$, with $j$ steps of size $\Delta = (P_{max} - P_{min})/j$.

The adaptive traffic probability at period $p' > p$ is then

$$P_t(q_i, p') = P_t(q_i, p) + T$$

$$T = \begin{cases} -\Delta & l_{n,q_i}(p') > l_{n,q_i}(p), P_t(q_i, p) \leqslant P_{min} \\ +\Delta & l_{n,q_i}(p') < l_{n,q_i}(p), P_t(q_i, p) \geqslant P_{max} \\ 0 & otherwise \end{cases}$$

This way, we reduce the probability of forwarding when message traffic increases, and vice versa.

### 3.1.2 Geocast with Limited Spread

To calculate the probability that another vehicle will forward the same message a vehicle checks how close it is to the edge of the angle of spreading $\theta$. Those closer to the center of the angle have an higher probability than those closer to the border, which can be seen in Figure 3.3 as the ratio of angles $\delta$ and $\theta$. To calculate $\delta$, we need a point $C$ inside the $DST\_AREA$, the location of the sender, $S = l_s(t)$ and the node's current location, $A = l_a(t)$. For every message received, we calculate the angle of incidence $\delta$ and the probability of forwarding $P_b(\delta, \theta)$ as follows:

$$\delta = acos(\overrightarrow{SA} \cdot \overrightarrow{SC} / |\overrightarrow{SA}||\overrightarrow{SC}|)$$

$$P_b(\delta, \theta) = \begin{cases} 1 - \delta/\theta & \delta \leqslant \theta \\ 0 & otherwise \end{cases}$$

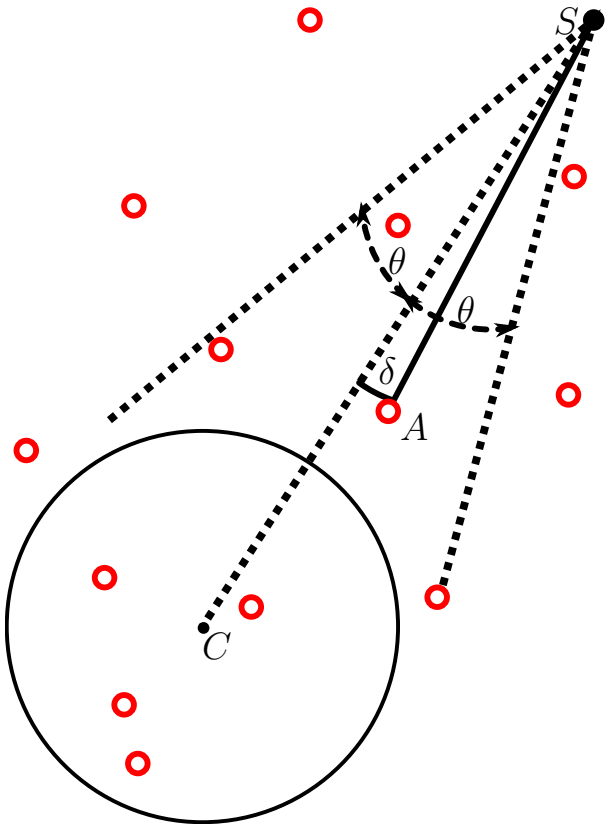


Figure 3.3: Probabilistic forwarding example.
A message is sent to $DST\_AREA$ with a spreading angle $\theta$.
Nodes with an angle of incidence $\delta \leqslant \theta$ will forward the message.

In other words, the closer a node is to the center of the triangle, the higher the probability of repeating the message. Nodes outside of the triangle discard the message.

Note that if a node $n_i$ were to receive the same message $m_i$ with the same value of $\theta$ from two distinct nodes $n_j, n_k$, the node would still have different probabilities of retransmission, since the senders' positions are different ($l_j(t) \neq l_k(t)$). Therefore, the value of $\delta$ is different for every incoming message. Moreover, even if $n_i$ were to receive the same message from node $n_j$ twice and their relative positions remained equal (i.e. $\overrightarrow{SA} \parallel \overrightarrow{S'A'}$), the value of $C$ is fixed, so $\overrightarrow{SC} \nparallel \overrightarrow{S'C}$, and the resulting $\delta$ would be different.

**Algorithm 1** Probabilistic routing algorithm operated by a node $n_i$

**function** FORWARD()($M_A, DST\_AREA, \theta, l_s(t), l_C$)
  Check authenticity of $M_A$
  **if** $M_A$ is authentic **then**
    $load \leftarrow sizeof(M_A)$
    $QueueProb \leftarrow$GetQueueProb($l_s(t), l_a(t), load$)
    $RouteProb \leftarrow$GetRouteProb($l_s(t), l_C, l_a(t), \theta$)
    $CombinedProb \leftarrow$Random($0, QueueProb \times RouteProb$)
    **if** ($CombinedProb > 0$) **then**
      **return** FORWARD
    **else**
      **return** DISCARD
    **end if**
  **end if**
**end function**
**function** GETQUEUEPROB(($l_s(t), l_a(t), load$))
  $n \leftarrow$DetermineQueue($l_s(t), l_a(t)$)
  $msgFreq(n,t) \leftarrow N_msg/(t - t_0)$
  $payload(n,t) \leftarrow payload(n) + load$
  $totaload(n,t) \leftarrow msgFreq(n) * k_1 + payload(n) * k_2$
  **if** $totalLoad(n,t) > totalLoad(n, t-1)$ **then**
    $P(t) \leftarrow MAX(P(t-1) - P_\Delta, P_{min})$ Where $P_\Delta$ is the step factor
  **else**
    $P(t) \leftarrow MIN(P(t-1) + P_\Delta, P_{max})$
  **end if**
  **return** $P(t)$
**end function**
**function** GETROUTEPROB(($l_s(t), l_C, l_a(t), \theta$))
  Let $SA \leftarrow \overrightarrow{l_s(t) l_a(t)}$
  Let $SC \leftarrow \overrightarrow{l_s(t) l_C}$
  $\delta = acos(SA \cdot SC / |SA||SC|)$
  $P(t) \leftarrow MAX(0, 1 - \delta/\theta)$
  **return** $P(t)$
**end function**

## 3.2 Performance Evaluation

### 3.2.1 Experimental Results

To gather empirical data, we used the *ns-3* network simulator with a VANET infrastructure. We are testing in a $4 \times 4km$ area during a simulation time of 60 seconds. We ran scenarios to see how the algorithm behaves at different levels of network density (500 - 1500 nodes), and we also see the effect that $\theta$ has over the communications. Algorithm 1 shows how the routing algorithm operates.

Intuitively, the algorithm will flood the network when $\theta = \pi$ and behave like line-of-sight communication with smaller values, such as when $\theta = \pi/180$. Since a larger value of $\theta$ creates a larger area where nodes within can participate in the routing, it also increases the probability of a message reaching the destination, at the cost of increased messaging due to flooding. Conversely, a lower value reduces network traffic but could result in lower reliability, especially in sparse networks. This is corroborated by the results shown in Table 3.2 and Figure 3.4.

### 3.2.2 Comparison with DSG

Since we approached the issue of privacy as our first goal, our implementation takes privacy measures not used in the DSG protocol and is therefore more secure. However, these privacy measures have a negative impact on network efficiency, especially when looking at the messaging overhead ratio.

Compared to the AAG protocol, we are not as efficient when comparing the ratio of wasted messages (see Table 3.1). Since AAG forwards the message directly to each node, it does not generate as many messages as our broadcast, in which the vast majority of nodes that receive a message will never participate in the routing (i.e. those outside

Table 3.1: Effect of node density over messaging overhead.

$I = 1, \theta = \pi/6$

| $N$ | $S$ | $F$ | $D$ | $\Sigma$ | $D/\Sigma$ |
|---|---|---|---|---|---|
| 500 | 11,710 | 2,723 | 48,677 | 63,110 | 0.7713 |
| 750 | 13,281 | 3,140 | 44,917 | 61,339 | 0.7323 |
| 1000 | 17,318 | 4,138 | 68,852 | 90,308 | 0.7624 |
| 1250 | 5,173 | 464 | 15,300 | 20,937 | 0.7308 |
| 1500 | 14,766 | 909 | 38,872 | 54,547 | 0.7126 |

Legend:
$N$ = Number of nodes. $F$ = Forwarded messages.
$S$ = Messages in $DST\_AREA$. $D$ = Discarded messages.

of the $\theta$ range) and immediately discard the message. Moreover, due to our efforts to protect privacy, we cannot match the connectivity and/or message efficiency of the DSG approach.

However, we consider our reliability levels and message efficiency to be manageable, and as we will show next, these levels can be adjusted to suit requirements. For certain values of $\theta$, we manage to reach connectivity levels above 90% (see Table 3.3).

#### 3.2.2.1 Node Density and Message Overhead

As previously mentioned, Table 3.1 shows that a large quantity of messages are not used in the routing process. The bulk of these messages are discarded due to being outside of the routing cone of the forwarding node.

Of interest, however, is the fact that network traffic ($\Sigma$) and the ratio of useful data ($1 - D/\Sigma$) are not affected by the node density of the network.This is because most of the discarded messages are at the border of the cone created by $\theta$.

Table 3.2: Effect of $\theta$ on messaging overhead.

$I = 1, n = 500$

| $\theta$ | $S$ | $F$ | $D$ | $\Sigma$ | $D/\Sigma$ |
|---|---|---|---|---|---|
| $\pi/24$ | 11 | 7 | 135 | 153 | 0.8824 |
| $\pi/12$ | 19 | 16 | 246 | 281 | 0.8754 |
| $\pi/8$ | 166 | 66 | 1,052 | 1,284 | 0.8193 |
| $\pi/6$ | 11,710 | 2,723 | 48,677 | 63,110 | 0.7713 |
| $\pi/4$ | 127,244 | 76,727 | 593,972 | 797,943 | 0.7444 |
| $\pi/3$ | 281,938 | 294,433 | 1,199,348 | 1,775,719 | 0.6754 |

Table 3.3: Load balance effect on throughput.

| | | Neighbouring initiators $I$ | | | | | |
|---|---|---|---|---|---|---|---|
| | $S$ | 5 | 10 | 15 | 20 | 25 | 50 |
| $\theta$ | $\pi/12$ | 2 | 2 | 2 | 4 | 4 | 3 |
| | $\pi/6$ | 3 | 4 | 5 | 5 | 7 | 8 |
| | $\pi/4$ | 4 | 7 | 4 | 3 | 11 | 7 |
| | $\pi/3$ | 5 | 9 | 5 | 9 | 9 | 12 |

Each item in the table represents the number of distinct messages that arrive at the destination area, with respect to the initiators and $\theta$.

#### 3.2.2.2 Angle of Spreading and Message Overhead

In Table 3.2 and Figure 3.4, we can see the effect $\theta$ has on the messaging overhead. Once again, while the number of wasted messages $D$ does indeed increase, the ratio of useful to wasteful messages $D/\Sigma$ is actually reduced when the value of $\theta$ increases. This reinforces the idea that the largest number of wasted messages is generated along the borders of the cone. This is more pronounced at smaller angles ($\theta \leqslant \pi/6$) and becomes less of a factor at larger angles. From empirical data, it seems that with a value of $\pi/12$, we achieve a good balance of spread and overhead.

#### 3.2.2.3 Load Balance Effect on Throughput

In Table 3.3, we show the effect traffic control has on the delivery success ratio, as well as the impact $\theta$ has. We had $I$ independent nodes sending a message to the same

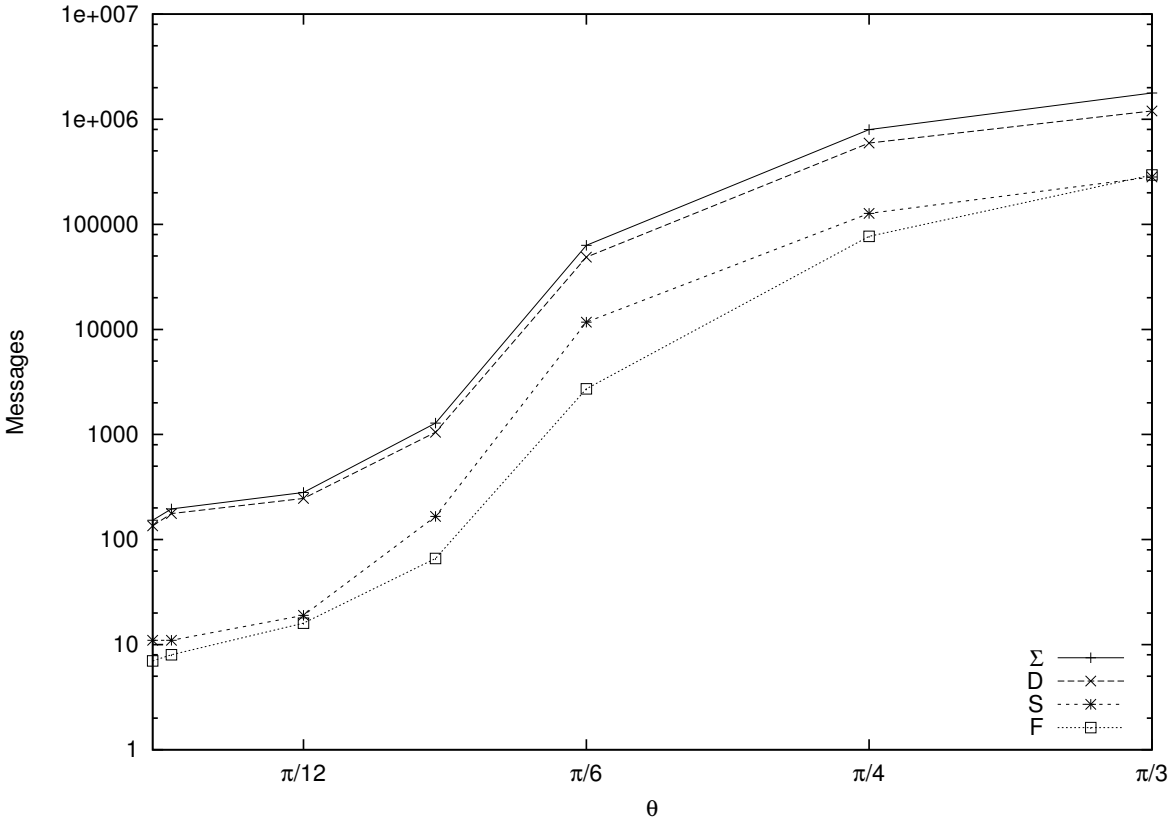


Figure 3.4: Relationship between $\theta$ and routing overhead.

$DST\_AREA$. The senders are all physically close to each other, are all using the same value of $\theta$ and are all sending the message simultaneously. This scenario (several nearby nodes simultaneously sending a message to the same location) is very likely caused by an event perceived by all cars in the vicinity (e.g. a car accident, road conditions, etc.) and messages are very likely to have duplicated content.

In this scenario, there is a great amount of overlap in the group of nodes that participate in the forwarding. Traffic restriction shows its effectiveness, limiting the number of messages that are forwarded to avoid network abuse. The data shows that $\theta$ has a clear impact on communication, with a visible correlation between the value of $\theta$ and the number of messages that arrive.

## 3.3 Conclusion

We propose a new scheme of private routing for VANETs, which uses a probabilistic forwarding algorithm. Our algorithm achieves security and privacy by not sending private information in the routing headers and by using pseudonymous identities. With this algorithm, we can also prevent flooding and blackhole attacks as well as reduce the efficiency of Sybil attacks. We will show how the algorithm preserves privacy and is effective in defending against security attacks in Chapter 5.

# Chapter 4

# Transmission-Coverage Algorithm

In this chapter, we present our transmission-coverage geocasting protocol. Once again, we first describe our geocasting protocol, and then provide empirical data to confirm our theories and decisions regarding routing.

## 4.1 The Protocol

When a vehicle sends a message, it sets the headers with values for routing and broadcasting purposes. The initial sender defines the $DST\_AREA$ and a point $C$ inside the area, as well as the arc of interest $\theta$ used for routing. While $C$ can be positioned anywhere inside the $DST\_AREA$, for the purpose of this paper we assume that it is in the center of the area. The $DST\_AREA$ itself can also be of any given shape and/or size. The values of the headers depend on the type of message the sender is producing.

For example, information could be sent about the fact that a road in area $A_{r+1}$ is to remain closed until 12 pm for maintenance, to enable vehicles in region $A_r$ to change their routes. The network could also be used to warn a hospital of a nearby accident so that it can dispatch an ambulance to the area. The sender position field is also set with

the value $l_i(t)$, so that receivers can decide whether to discard the message or forward it after a certain delay. If the sender is between the receiver and the target area, there is little incentive for the receiver to forward the message, since it is farther from the destination than the sender itself.

The message header fields are identical to those used in the probabilistic algorithm seen in Chapter 3, where Figure 3.1 offers a visual reference. The main difference is the meaning applied to the value of the field $\theta$ and its use in the routing algorithm.

- *$DST\_AREA$:* The geographic area describing the destination location (e.g centre and radius).
- *Angle of interest $\theta$*: Sets the arc of interest, which is used when routing the message outside the $DST\_AREA$.
- *Timestamp, Expiration Time* and *TTL:* Used in tandem to measure the validity of the message with respect to time and number of hops. A message that exceeds the validity time window and/or number of hops is discarded.
- *Sender Position:* When a node decides to forward a message, it updates this field with its current location, $l_i(t)$. A receiving node checks this field to determine the coverage area that is common between the sender and the receiver, and adjusts the transmission delay accordingly.

After all the headers are generated, the message is encrypted and signed. The data packet is then broadcast omnidirectionally.

Upon receipt of a message, a node checks its validity (timestamp, signature, TTL, etc.) to discard invalid messages. If the message is valid, it is then considered for retransmission. Instead of deciding whether to discard or forward the message immediately, a receiver checks the transmission area that has been covered by other transmissions and

delays sending the message for a discrete period of time. During this delay, if the node receives duplicates of the message, it updates the value of the transmission coverage left, and increases or resets the amount of time it waits to avoid broadcasting a message that is redundant. The duration of the transmission delay is calculated using variables other than just area coverage. The location and number of requests to forward the message in question also affect the delay.

Moreover, the decision to forward a message varies depending on whether the vehicle is inside the geocast area or outside of it. Nodes inside the geocast area attempt to broadcast the message to as many neighbors as possible, while the main objective of transmission outside the area is to deliver the message to the destination in a reliable fashion.

To achieve this result, the nodes inside the destination area use shorter delays before broadcasting when they can reach as many nodes as possible with one broadcast, creating the least amount of messaging. Nodes outside the destination area also accelerate the rate of retransmission when nearing the destination by reducing the lengths of the delays.

Regardless of the action taken (routing or broadcasting), the message headers are filled appropriately. That is to say, even inside the $DST\_AREA$, the value of $\theta$ is still left in the headers – even if it is of no use – to maintain a uniform packet size.

### 4.1.1 Inside the Geocast Area

When a vehicle inside the geocast area receives a message, it applies the transmission-coverage algorithm to determine whether or not to retransmit the message and if applicable the delay it should apply to the transmission. The reason for this delay is to avoid sending a duplicate message to a vehicle or area that has previously received the message.

The delay is based on three elements: *i)* the transmission area of the vehicle that has not received the message, *ii)* the distance between the vehicle and the center of the destination area, and *iii)* a random value used to resolve ties between vehicle retransmissions.

#### 4.1.1.1 Distance to Center

When a node receives a copy of the message, it determines how close it is to the center of the destination area. To ensure that the message arrives in a timely fashion, the closer a node is to the destination, the less it will wait to retransmit the message. Even when the message has reached the center $C$, transmission continues until the $DST\_AREA$ is fully covered. In Figure 4.1, since vehicle $A$ is closer to the destination center $C$, it is going to retransmit the message before vehicle $B$. That is, if $t_A$ is the delay time that node $A$ sets to retransmit the message, it will be shorter than the time $t_B$ that node $B$ will set to retransmit it. Essentially, $t_A \propto |\overrightarrow{CA}|$

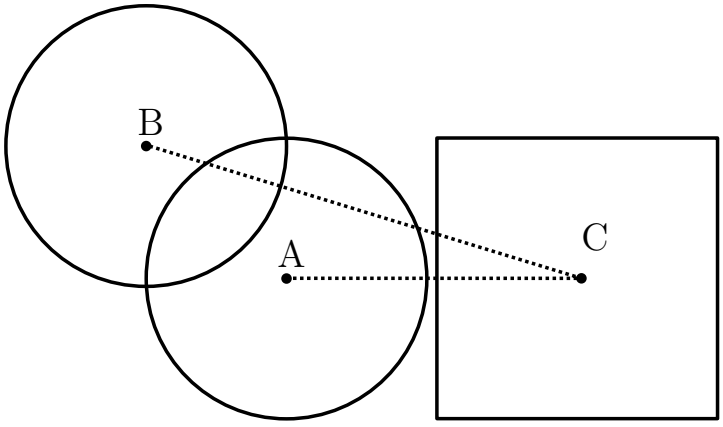


Figure 4.1: Relationship between distance and delay.
Since vehicle $A$ is closer to $C$, it will forward the message before $B$.

#### 4.1.1.2 Transmission Area

Every vehicle has a transmission range that covers a circular area of radius $r$ around them, with the vehicle in the center of the transmission area. When a node transmits a message, it can assume that every node inside the transmission range (i.e. less than a distance $r$ away) has received a copy of the message. Likewise, when a node receives a message from another node, it knows that in the transmission range of that node's transmission area, all the nodes have received the message, the receiving vehicle included. The receiving node can also calculate the area where the transmission of both the sender nodes and its own overlap, and is aware that all the nodes inside it have received a copy of the message. We call this overlapping area the *covered area*. If the receiving vehicle were to broadcast the message, it would only be useful to those nodes that are outside of the overlapping transmission area (i.e. the uncovered area). For those nodes that are in the covered area, the retransmission done by this node would only give them a duplicate of the message. Following this logic, when the transmission area of a node is fully covered, all of the nodes it can transmit the message to have already received a copy of it; broadcasting the message would result in a needless duplication of the message.

To avoid sending a duplicated message, every time a copy of the message is received, the node calculates the covered transmission area, $A_C$ and updates the transmission delay time, $t$. Since a larger $A_C$ translates to a smaller uncovered area, which is the area that benefits from this node's broadcast, the larger $A_C$ is, the less useful a broadcast is, so it can be further delayed. For that reason, a larger $A_C$ implies a larger value of $t$. In other words, $t \propto A_C$.

If the transmission area of the node is fully covered by other nodes, it decides that it is not necessary to forward the message (this is done by setting $t = \infty$). When the delay has elapsed, the node forwards the message only if there is an area within its

transmission range that has not received a copy of the message from another source. For a more detailed explanation of how the transmission area coverage is calculated, see Section 4.1.3.

#### 4.1.1.3 Random Value

In the event that two vehicles are within the same zone and have the same coverage, a random value, $R$, is used to avoid creating a tie. The benefit of minimizing ties is that if both vehicles are within range of each other, using the logic above, when a node receives the duplicated message from a neighbour, it increases the delay to retransmit it, and it is possible that it can forgo transmitting it altogether, thus minimizing the number of duplicate messages generated.

Combining these values, we get

$$t = |CA|A_C R \tag{4.1}$$

### 4.1.2 Outside the Geocast Area

Outside of the geocast area, the reason behind transmission is to route the message instead of broadcasting it, so the algorithm is slightly modified to take advantage of this.

In the routing zone, the message needs to arrive in the geocast area as quickly as possible. Since it is not necessary to ensure that all the nodes in the routing zone receive the message, nodes forward the message only when it moves the message closer to the destination area. To determine when the message advances, the node sets a section of the perimeter that faces the destination area, and is $2\theta r$ in length, where the value of $\theta$ is provided in the message header. When this arc is covered by a transmission from a neighbouring vehicle, it follows that the vehicle that sent the message is closer to the

$DST\_AREA$ than the receiving vehicle, so retransmitting the message is of very little value. Therefore, the receiver vehicle is not required to forward the message any longer, even if the rest of the perimeter has not been fully covered.

### 4.1.3 Transmission Area Coverage

Upon receipt of a message $M$, the vehicle calculates the area where the transmission of the sender and its own transmission intersect. To do this, it uses the sender's position $S$, as well as its own position $A$, and determines the points $P_1, P_2$ that both transmission perimeters have in common to obtain the area that is covered.

The vehicle then calculates the distance $d$ between the two vehicles, as well as the distance $h$, which is the length of the line that is perpendicular to $\overrightarrow{SA}$ and passes over the points $P_1, P_2$.

Once the points are obtained, the source calculates the angle $\sigma$ formed by $P_1AP_2$, and calculates the circular segment area $C_a$, which is the area where the two circles overlap. See Figure 4.2 for a graphic example of this process.

To determine $P_1, P_2$ and $L_a$, the vehicle does the following calculation.

$$\begin{aligned}
\text{Let } V_\Sigma &= (A+S)/2 \\
V_\Delta &= A - S \\
d &= |V_\Delta| \\
h^2 &= r^2 - (d/2)^2 \\
\text{Then } P1(x) &= V_\Sigma(x) + h/d * V_\Delta(y) \\
P2(x) &= V_\Sigma(x) - h/d * V_\Delta(y) \\
P1(x) &= V_\Sigma(y) - h/d * V_\Delta(x) \\
P2(x) &= V_\Sigma(y) + h/d * V_\Delta(x) \\
cos\sigma &= \frac{\overrightarrow{|P_1A|}^2 + \overrightarrow{|P_2A|}^2 - \overrightarrow{|P_1P_2|}^2}{2\overrightarrow{|P_1A|}\overrightarrow{|P_2A|}} \\
C_a &= r^2(\sigma - sin\sigma)
\end{aligned}$$

Given this information, the receiving vehicle can determine the area in its transmission range that presumably has also received the message and therefore does not need to receive a duplicate of the message. The complement of this is the area that has not yet received the message. As the area that remains to be covered is reduced, so is the probability that there is a node in that area that needs to receive the message. Therefore, nodes that have the least amount of area to cover also have the least urgency to repeat the message, while nodes that have a larger area left to cover can cover the greatest number of nodes with one retransmission and thus decide to retransmit faster.

To accomplish this, every time the node receives a duplicate of the message, it calculates the percentage of area covered and updates the transmission delay accordingly.

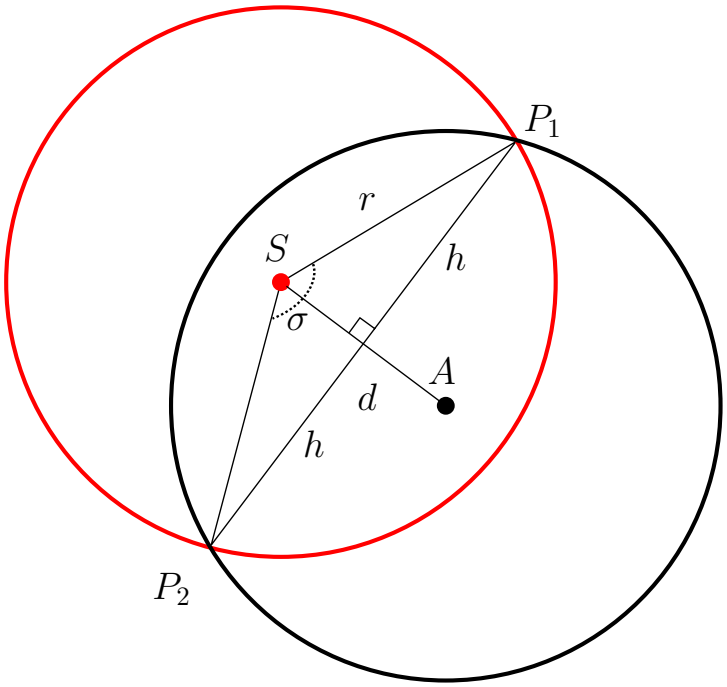


Figure 4.2: Transmission coverage overlap.
The intersection between nodes $S$ and $A$ is used to determine coverage.

The percentage can be calculated in the following form:

$$\begin{aligned} Coverage &= \frac{C_a}{\pi r^2} \\ &= \frac{r^2(\sigma - sin\sigma)}{\pi r^2} \\ &= \frac{\sigma}{\pi} - \frac{sin\sigma}{\pi} \end{aligned}$$

Once the entire transmission area is covered (i.e. $C_a = \pi r^2$), every neighbouring node that is within the transmission radius has already received a copy of the message from another source. Broadcasting the message is then unnecessary, as it only serves to create a duplicate. For this reason, the forwarding node decides to not forward the message.

Since the radius of transmission of both vehicles is the same, the values of $\sigma$ are always small. Therefore, for the purpose of our algorithm, the ratio of arc length covered

by a transmission with respect to coverage perimeter is proportional to the area covered by a transmission with respect to coverage area.

If we let $L_a$ be the covered arc length, we then have:

$$\begin{aligned} L_a &= r\sigma \\ Coverage &= \frac{L_a}{2\pi r} \\ &= \frac{r\sigma}{2\pi r} \\ &= \frac{\sigma}{2\pi} \end{aligned}$$

The benefit of using the perimeter to calculate the coverage area is that it simplifies the calculation when there are overlapping transmissions. We can treat the perimeter as a line segment of length with range $[0, 2\pi r]$ and each subsequent transmission covering a perimeter $l_i$ of length $|l_i|$ as a line segment with range $[p, p + |l_i|]$. When two different retransmissions cover the perimeters $l_1, l_2$, it can be modeled as line segments $[p_1, p_1 + |l_1|], [p_2, p_2 + |l_2|]$ respectively, and it is simple to check if there is overlap.

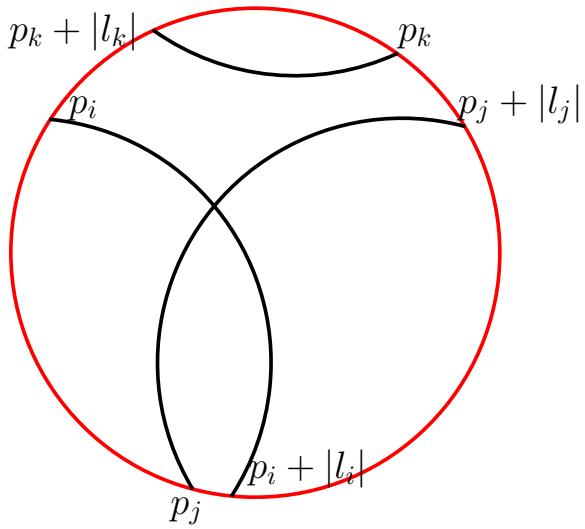


Figure 4.3: Overlapping perimeters.

Namely, perimeter $l_2$ overlaps with $l_1$ when $l_1 \cap l_2 \neq \emptyset$, which is defined as follows:

$$l_1 \cap l_2 \neq \emptyset \iff \{p_1, p_1 + |l_1|\} \in [p_2, p_2 + |l_2|] \wedge \{p_2, p_2 + |l_2|\} \in [p_1, p_1 + |l_1|]$$

We have: $L_a = \bigcup_i l_i$

Where $\forall i, j \; i \neq j$

$$l_i \cap l_j \neq \emptyset \rightarrow |l_i \cup l_j| = |[min(p_i, p_j), max(p_i + l_i, p_j + l_j)]|$$

$$l_i \cap l_j = \emptyset \rightarrow |l_i \cup l_j| = |l_i| + |l_j|$$

This way, when $l_i$ and $l_j$ overlap, they are merged together, while being kept as separate parameters when they do not. Figure 4.3 shows how two perimeters might intersect.

## 4.2 Performance Evaluation

### 4.2.1 Experimental Results

To gather empirical data, we used the *ns-3* network simulator with a VANET network infrastructure. Algorithm 2 shows how the routing algorithm operates.

We ran the simulations in a $4 \times 4km$ area for a period of $60s$ of network operation. We ran scenarios to see how the algorithm behaves at different levels of network density (500 - 1500 nodes), and we also saw the effect that $\theta$ has over the communications. Experimentally, we found that the transmission radius of the vehicles in the network was around $120m$. While more powerful or accurate devices could be conceived, we believe that this transmission radius is effective for the messaging operations at hand. Moreover, we assume that there are no obstructions to transmissions that could constrain or reduce this transmission range.

**Algorithm 2** Deterministic routing algorithm operated by a node $n_i$

1: **function** FORWARD( $(M_A, DST\_AREA, \theta, l_s(t), l_a(t), l_C)$)
2: Check authenticity of $M_A$
3: **if** $M_A$ is authentic **then**
4: **if** $(l_a(t) \in DST\_AREA$ ) **then**
5: $coveredArea \leftarrow$CalcCoveredArea$(l_s(t), l_a(t), l_C, 2\pi)$
6: **else**
7: $coveredArea \leftarrow$CalcCoveredArea$(l_s(t), l_a(t), l_C, \theta)$
8: **end if**
9: **if** $(coveredArea = 1.0)||(M_A$ was sent before) **then**
10: **return** DISCARD
11: **else**
12: $delayTime \leftarrow CalcDelayTime(l_s(t), l_a(t), l_C, coveredArea)$
13: **if** Message previously scheduled to be forwarded **then**
14: $ReScheduleForward(M_A, delayTime)$
15: **else**
16: $ScheduleForward(M_A, delayTime)$
17: **end if**
18: **end if**
19: **end if**
20: **end function**
21: **function** CALCCOVEREDAREA$((l_s(t), l_a(t), l_C, \theta))$
22: $arcCovered_\theta \leftarrow$CalcThetaCovered$(2\pi - \theta, l_a(t), l_C)$
23: $arcCovered \leftarrow$CalculateArcIntersection$(l_s(t), l_a(t), r)$
24: $Covered \leftarrow Covered \cup arcCovered \cup arcCovered_\theta$
25: $P(t) \leftarrow MIN(Covered/2\pi r, 1)$
26: **return** $P(t)$
27: **end function**
28: **function** CALCDELAYTIME$((l_s(t), l_a(t), l_C, coveredArea))$
29: **if** $coveredArea = 1$ **then**
30: $delay \leftarrow \infty$
31: **else**
32: Let $dist \leftarrow |\overrightarrow{l_a(t)l_C}|$
33: $rand \leftarrow Random()$
34: $delay \leftarrow dist * coveredArea * rand$
35: **end if**
36: **return** $delay$
37: **end function**

Unlike the probabilistic modes, this algorithm is influenced by the number of nodes in the network. This is because with higher node densities, nodes are closer to each other. Therefore, when a node transmits a message, it will cover a larger part of the neighbouring nodes' transmission area, increasing their probability of being fully covered and deciding not to transmit. Therefore, node density subtly shapes network traffic and levels.

When routing, a larger value of $\theta$ translates to a larger uncovered transmission area, which decreases the initial transmission delay and would ostensibly increase the number of forwarding nodes and/or routing paths. Concretely, if the value of $\theta = \pi$, the algorithm behaves the same outside the $DST\_AREA$ as inside, while a small value like $\theta = \pi/180$ would result in nodes taking longer to retransmit messages, since the initial uncovered area is smaller than normal. Table 4.2 shows that, indeed, the number of messages used to calculate the area coverage increases, yet the number of messages forwarded does not seem to be affected by different values of $\theta$.

#### 4.2.1.1 Node Density and Message Overhead

As suggested, Table 4.1 shows that a large number of messages are duplicates not used in the forwarding process.

The bulk of these messages, however, are discarded when the node realizes that its transmission range has been fully covered already (shown in the column $D_{full}$). Moreover, many duplicated messages are used for calculating the transmission area coverage (column $E$). This, combined with the fact that the number of forwarding operations $F$ does not fluctuate as much as the rest of the messages and remains relatively small compared to the total number of messages $\Sigma$, strongly supports the theory that not all nodes need to forward the message upon receipt.

Table 4.1: Effect of node density on messaging overhead.

$I = 1, \theta = \pi/6$

| $N$ | $S$ | $F$ | $D$ | $D_{full}$ | $E$ | $\Sigma$ | $(D+E)/\Sigma$ |
|---|---|---|---|---|---|---|---|
| 500 | 49 | 92 | 1,104 | 945 | 922 | 2,167 | 0.9349 |
| 750 | 65 | 117 | 2,621 | 2,400 | 1,811 | 4,614 | 0.9606 |
| 1000 | 143 | 118 | 3,398 | 3,164 | 2,538 | 6,197 | 0.9579 |
| 1250 | 154 | 110 | 4,357 | 4,113 | 3,021 | 7,642 | 0.9655 |
| 1500 | 212 | 97 | 4,890 | 4,711 | 3,332 | 8,531 | 0.9638 |

Legend:
$N$ = Number of nodes. $F$ = Forwarded messages.
$S$ = Messages in $DST\_AREA$. $D$ = Total Discarded messages.
$D_{full}$ = Message discarded when transmission area is fully covered.
$D_{full} \in D$ $E$ = Messages that affect coverage
$\Sigma = S + F + D + E$

Total network traffic ($\Sigma$) is affected by the node density of the network, with higher node densities having more dropped messages per forwarded message. However, the percentage of discarded data ($(D + E)/\Sigma$) remains stable at all densities. The reason behind this behaviour is that at higher densities, the transmission of a message affects more nodes. This in turn causes a larger number of messages ($E$) that modify the transmission coverage upon receipt.

#### 4.2.1.2 Angle of Coverage and Message Overhead

In Table 4.2 and Figure 4.4, we can see the effect $\theta$ has on the messaging overhead. As expected, a smaller value of $\theta$ translates into less coverage area for the routing nodes, and results in a smaller number of messages used for calculating the transmission area ($E$), while at the same time increasing the number of messages discarded upon receipt because the transmission area is fully covered ($D_{full}$) faster.

The other reason a node refuses to discard a message is because it receives a duplicate of a message it already forwarded. The number of messages being discarded for this

Table 4.2: Effect of $\theta$ on messaging overhead.

$I = 1, n = 500$

| $\theta$ | $S$ | $F$ | $D$ | $D_{full}$ | $E$ | $\Sigma$ | $(D+E)/\Sigma$ |
|---|---|---|---|---|---|---|---|
| $\pi/24$ | 76 | 94 | 1,220 | 1,084 | 887 | 2,277 | 0.9253 |
| $\pi/12$ | 47 | 99 | 1,241 | 1,086 | 928 | 2,315 | 0.9369 |
| $\pi/8$ | 58 | 101 | 1,143 | 991 | 1,023 | 2,325 | 0.9316 |
| $\pi/6$ | 49 | 92 | 1,104 | 945 | 922 | 2,167 | 0.9349 |
| $\pi/4$ | 71 | 125 | 1,231 | 924 | 1,260 | 2,687 | 0.9271 |
| $\pi/3$ | 81 | 109 | 1,178 | 835 | 1,344 | 2,712 | 0.9299 |

reason $(D - D_{full})$ grows as $\theta$ increases. Interestingly, the outcome of this is that the network load ($\Sigma$) does not seem to be affected by the value of $\theta$.

Furthermore, the arrival rate $A$ and the number of forwards $F$ do not seem to be affected by the value of $\theta$, which goes against the original expectations. For these reasons, it is logical to conclude that the value of $\theta$ does not ultimately affect network traffic.

Table 4.3: Distinct messages inside $DST\_AREA$ ($S$).

| | | $I$ | | | | | |
|---|---|---|---|---|---|---|---|
| | $S$ | 5 | 10 | 15 | 20 | 25 | 50 |
| $\theta$ | $\pi/12$ | 5 | 8 | 13 | 16 | 17 | 31 |
| | $\pi/8$ | 5 | 9 | 13 | 16 | 17 | 34 |
| | $\pi/6$ | 4 | 8 | 11 | 15 | 15 | 30 |
| | $\pi/4$ | 5 | 8 | 9 | 13 | 15 | 22 |
| | $\pi/3$ | 5 | 8 | 8 | 9 | 15 | 21 |

Table 4.4: Distinct messages outside $DST\_AREA$ ($F$).

| | | $I$ | | | | | |
|---|---|---|---|---|---|---|---|
| | $F$ | 5 | 10 | 15 | 20 | 25 | 50 |
| $\theta$ | $\pi/12$ | 5 | 9 | 13 | 17 | 21 | 37 |
| | $\pi/8$ | 5 | 9 | 13 | 16 | 19 | 36 |
| | $\pi/6$ | 5 | 9 | 12 | 15 | 18 | 33 |
| | $\pi/4$ | 5 | 9 | 11 | 13 | 16 | 27 |
| | $\pi/3$ | 5 | 9 | 11 | 13 | 16 | 24 |

Each item in the table represents the number of *distinct* messages that arrive in the destination area, with respect to the initiators and $\theta$. For example, when $\theta = \pi/6$, and 15 senders create a message, 11 of those 15 messages arrive at a node inside the $DST\_AREA$, and the message is forwarded 12 times.

Table 4.3 shows the number of distinct messages that arrived in the $DST\_AREA$, while Table 4.4 shows the number of messages that were seen by at least two nodes outside the $DST\_AREA$. We had $I$ independent nodes sending a message to the same $DST\_AREA$. All the senders have the same parameters ($DST\_AREA, \theta$ and initial sending time) and are all randomly placed on the network.

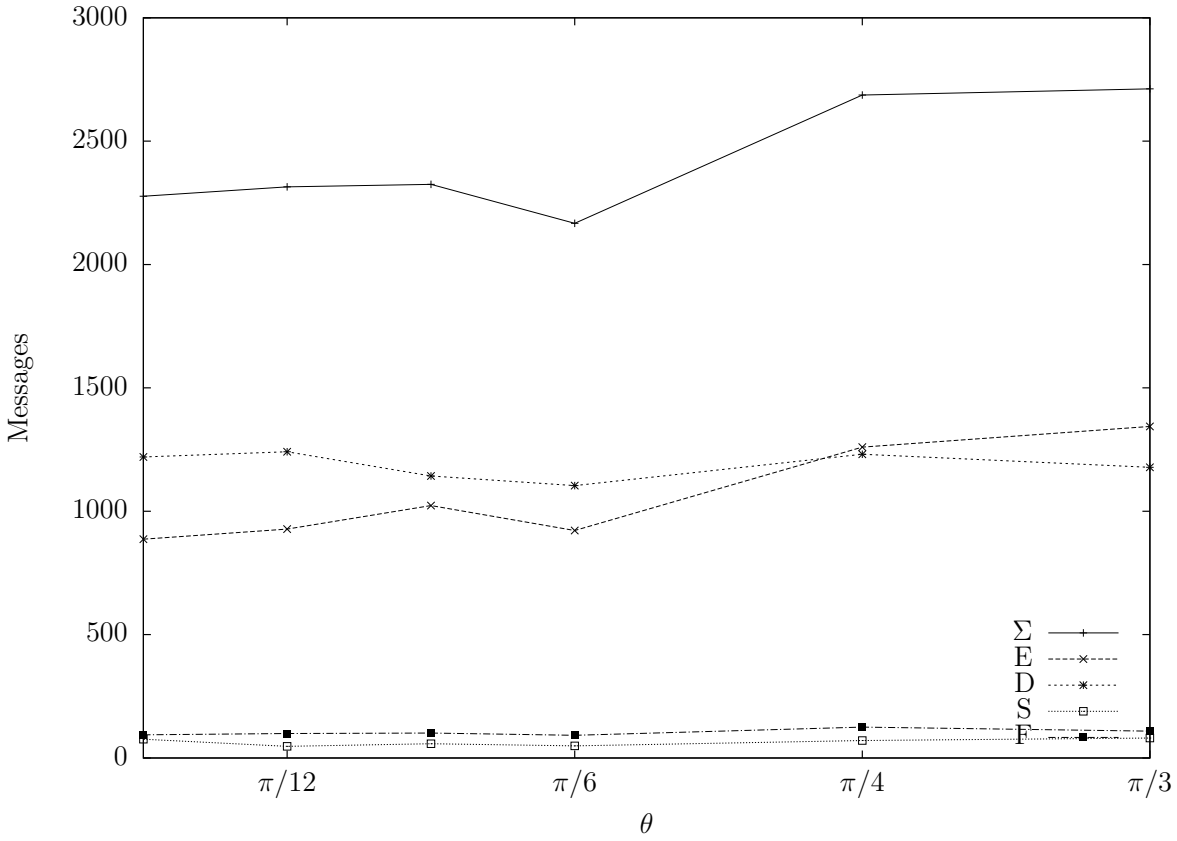


Figure 4.4: Relationship between $\theta$ and routing overhead.

In Table 4.3, we show the reliability of the algorithm, as well as the impact $\theta$ has on this property. We see that the larger $\theta$ is, the less reliable the algorithm becomes. We know that the only time $\theta$ affects the behaviour of the protocol is during the routing phase.

From Table 4.2, we see that higher values lead to longer delays, as suggested by the rising number of $E$ messages being used to calculate coverage. Every time the coverage updates (i.e. after receiving an $E$ message), the timer is reset.

Since the simulation collects data for a period of $60s$, it is possible that the rest of the messages do arrive later. On the other hand, the random positioning of the nodes also results in the senders being disconnected from the network. Table 4.4 shows that not all

messages are forwarded, a result of the senders being out of the transmission range of the other nodes in the network.

## 4.3 Conclusion

We propose a new scheme of private routing for VANETs, which uses a transmission-coverage forwarding algorithm. With this algorithm, we see a much-improved level of reliability, as well as a reduced level of duplicated messages. Our algorithm achieves security and privacy by not sending private information in the routing headers and by using pseudonymous identities. We will show that the algorithm preserves privacy in Chapter 5.

# Chapter 5

# Security Algorithm

## 5.1 The Algorithm

To securely transmit information, we encrypt the message in such a way that only intended recipients belonging to a particular geographic area can decode it. Secure geocasting is an access-control technique in which we are trying to limit access to nodes belonging to a particular region. Please note that our problem is much simpler than the attribute-based access control technique of Sahai and Waters (2005) and hence has not been discussed here. The encryption proceeds as follows:

Let $PK/SK$ be the public/private key pair for a pseudonym. The state transport authority can behave as the Key Distribution Center (KDC), which will be responsible of managing the secret credentials of vehicles. The KDC chooses $a_1, a_2, \ldots, a_m \in \mathbb{Z}_q$ randomly for each zone $A_1, A_2, \ldots, A_m$. These are kept secret – known only to the KDC – and are refreshed from time to time.

A secret $y \in \mathbb{Z}_q$ is also chosen by the KDC and refreshed from time to time. The KDC securely chooses a group $G$ of order $q$ and a generator $g$. It then calculates $g^{a_1}, g^{a_2}, \ldots, g^{a_n}$. With $G$, the set $Y = e(g, g)^y$ is evaluated from time to time.

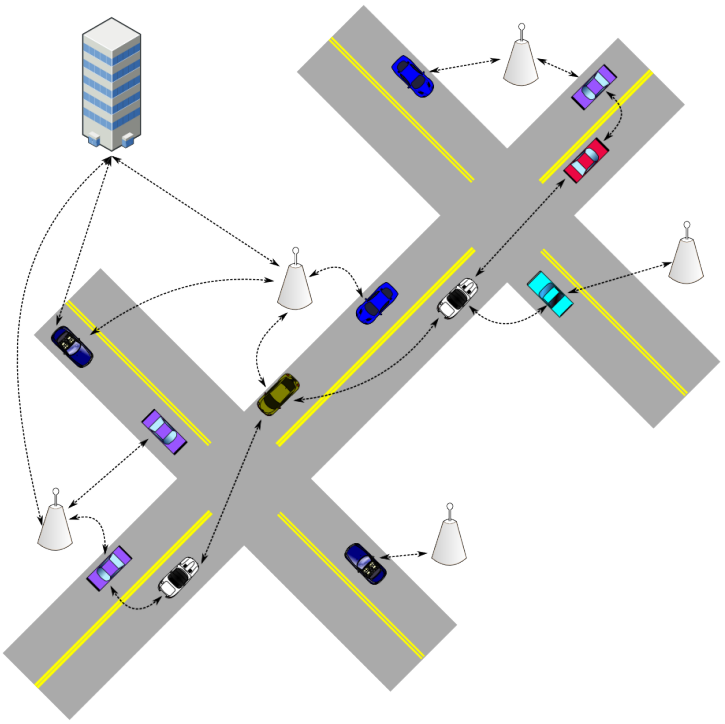

Figure 5.1: Communication using certificates.
The KDC generates and delivers the certificates to the RSU's and the vehicles.
Vehicles then communicate with RSUs to obtain a group key.

The KDC provides the RSU in the geographic location $A_r$ with the secret key $g^{y/a_r}$. The secret parameters ($SPM$) and the public parameters ($PPM$) of the protocols are then:

$$SPM = \langle a_1, a_2, \ldots, a_m, y \rangle$$

$$PPM = \langle g^{a_1}, g^{a_2}, \ldots, g^{a_m}, Y \rangle$$

The public parameters $PKM$ are sent by the nearest RSU when a vehicle enters a geographical location. By presenting its pseudonymous identity to an RSU in region $A_r$, the vehicle also receives a geographic group secret key $g^{y/a_r}$ that is encrypted by the public key ($PK$) of the vehicle. This prevents all vehicles with invalid credentials from

reading the key $g^{y/a_r}$ and thus blocks them from decrypting information they are not supposed to receive. Encryption and decryption proceed as follows:

1. Encryption: A vehicle wishing to send information to a specific geographic location $A_r$ does the following:

   - Chooses $s \in \mathbb{Z}_q$ randomly
   - Calculates $Y^s$
   - Calculates $C_0 = g^{a_r s}$ from $g^{a_r}$
   - Sends ciphertext $C = \langle m_i Y^s, C_0 \rangle$.

2. Decryption: Upon receiving a message, a node calculates the following

$$e(C_0, g^{y/a_i}) = e(g^{a_r s}, g^{y/a_i}) = \begin{cases} NULL, i \neq r \\ e(g,g)^{ys}, i = r \end{cases}$$

   From $m_i Y^s$ and $Y^s$, message $m_i$ can be calculated.

The problem with this approach is that any intermediary node can change the message and pretend to be a valid sender. Therefore, we need to authenticate the source node that has sent the message.

### 5.1.1 Authentication

The encrypted message $C$, along with the headers containing $DST\_AREA$, Timestamp, Expiration time and $\theta$ is signed by the sender. This ensures that intermediary nodes are not able to modify any of these parameters.

The concatenated message $m = C||DST\_AREA\ ||Timestamp||Expiration\_time||\theta$ is signed using the protocol in 2.6. The signature

$$\sigma = H(m)^x \in G'$$

is calculated using Equation (2.1) where $x$ is the secret key chosen for the sender. The sender then broadcasts $sig = \langle C, \sigma, cert \rangle$ along with the header. This is the protected message, and it is padded with zeros so that the length of all messages is the same.

When a valid node receives the information $sig$, it verifies the certificate $cert$. Signature $\sigma$ is then checked using Equation (2.2). If neither the message nor the other fields are modified, the verification proceeds smoothly.

### 5.1.2 Overhead Costs

We now show the computation and communication overheads for message traversal, encryption and decryption, as well as sender authentication.

- Computational costs:

  During encryption, there are two exponentiation operations ($Y^s$ and $g^{a_r}$) while decryption involves one pairing operation. Using the Pairing Based Cryptography library (PBC) (Lynn, 2007) on a 32-bit 3GHz Pentium IV machine with an MNT curve of embedding degree $k = 6$ and $q = 160$ bit curve, the time for exponentiation is $T_{exp} = 0.6ms$ and the time to compute pairings is $T_p = 4.5ms$ (Devegili et al., 2007). Thus, encryption has a computation overhead of $1.2ms$ for every message encrypted and decryption takes 4.5 ms for every message decrypted.

  During authentication, the signing procedure involves one hash computation and exponentiation, which requires $0.6ms$ to compute.

For the verification procedure, two pairing operations are used to obtain $e(\sigma, g)$ and $e(H(C), X)$, which require $9ms$ to calculate.

- Message Size:

  To ensure message integrity, the $g^{a_r}$ needs to be sent which takes $\log |G|$ bits where $|G|$ is the size of the group $G$. Communication overhead is $\log |G'| + size(cert)$ to send the certificate and signature.

## 5.2 Security of our Algorithms

Now we show that both the direction-based and the transmission-coverage algorithms shown previously are secure and preserve privacy. We will first show that an insider attacker cannot read a message that it is not supposed to read, and neither can it pretend to be the sender and change the information.

**Theorem 1** *Our protocols are secure: a node cannot decrypt a message that it is not supposed to, and intermediary nodes cannot change the content of the message without being detected.*

**Proof:**

If a vehicle is in a region $A_i \neq A_r$ ($A_r$ is the target geographic region), then it is not possible to decrypt the message sent to region $A_r$. We note that it is computationally difficult to calculate $a$ given $g^a$ (Discrete Logarithm Problem). So a vehicle cannot calculate $a_1, a_2, \ldots, a_m$ or $y$ given $g^{a_1}, g^{a_2}, \ldots, g^{a_m}$, and $Y$. This is because it does not have the secret key $g^{y/a_r}$. Thus, $e(C_0, g^{y/a_i}) = NULL$, since $i \neq r$. $Y^s$ cannot be calculated and hence $m_i$ cannot be calculated. A node that has left region $A_r$ and still has the secret key $g^{y/a_r}$, can decrypt all messages to $A_r$ for a certain time. However,

due to key refreshment, the values of $y$ and $a_r$ change. The node is therefore unable to calculate any messages in the future.

Our signature scheme guarantees that a vehicle that receives the ciphertext cannot change it and pretend to be the source. The ciphertext is signed using signature $\sigma$ and sent along with the certificate of the source node. The source node chooses a random number $x$ and calculates $H(C)^x$. It also sends the certificate of the public key $X = f^x$. An intermediary vehicle cannot calculate $x$ from $f^x$, so it cannot modify the ciphertext and send a new message $sig' = \langle C', \sigma' = H(C')^x, cert \rangle$. It also cannot sign a randomly generated message and send it with a new certificate, because a valid certificate from the trusted authority (TA) should exist. Therefore, an adversary cannot change the content of the message without being detected.

As a result, our protocols are secure. □

**Theorem 2** *Our protocols preserve privacy.*

**Proof:**

Since both routing algorithms use the same headers for transmission and deliver the same data, it follows that if one is secure the other is as well. We can then prove that the protocols preserve privacy by showing that communications are pseudonymous and unlinkable, and that message traffic does not generate any useful information regarding privacy.

Since only the CA knows the unique ID of a vehicle, with a sufficiently large number of pseudonyms issued an adversary cannot trace the pseudonyms to the unique identity of the sender, making the communications pseudonymous. The only routing data that is unique for each vehicle is the current location at the time of forwarding. Since it is not sent along any other vehicle-specific information (e.g. ID of sender vehicle, ID of destination vehicle, etc.), there is no secondary parameter that can be used to link a

message with its forwarding node. This idea, combined with the fact that the destination is also not a specific node, but rather a geographical location, deters the adversary from detecting a link between endpoints.

In the direction-based algorithm, an adversary cannot identify a vehicle by the message forwarding pattern it uses, since every node makes a forwarding decision based on a combination of probabilistic forwarding and dynamic traffic restriction. The probability of forwarding is calculated using $\delta$, which as we previously mentioned changes even when the message is sent repeatedly from a source node.

An adversary is not aware of the number of groups $q_i(n)$ that an attacked node may have, so it cannot predict the overall effect it could have on an individual node. If an attacker wants to affect the traffic restrictions by saturating one of the node groups $q_i(n)$, the message rate and payload are also affected by the actions of other nodes in the group. Moreover, the threshold values of the group ensure that after a certain point, a change in network traffic is not going to affect forwarding probability, so non-colluding adversaries are unable to control the forwarding rate of any group $q_i(n)$.

As for the transmission-coverage algorithm, the reaction to receiving a message is to delay transmission. With sufficient coverage a node will not transmit a message. Moreover, since there are no message acknowledgements, curious vehicles cannot incite a vehicle to forward a message in a specific pattern. Since the transmission delay algorithm uses a random variable, it is not possible to determine network or vehicle information by merely tracking the time between messages being sent from a particular position.

Therefore, adversaries in the network have no way of obtaining private information about a vehicle based on message source or destination, and they cannot make any decisions based on messaging traffic information, such as transmission patterns. The communications are pseudonymous and unlinkable, and privacy is preserved. □

**Theorem 3** *The direction-based algorithm is secure against Sybil attacks.*

**Proof:**

Since a node is always grouped with other nodes, it would seem like an ideal place to mount a Sybil attack to flood the network with messages, but grouping renders an attack useless. An individual attacker cannot overload its given group when it is already saturated from regular traffic. If it were to send messages under several identities, they would nevertheless all be treated under the same group. The receiving node reacts the same way as it would had all the messages originated from just one identity; it updates the traffic value and is still bounded by the threshold levels.

Moreover, if the attacker tries to flood the network by pretending to be in other groups (by changing the value of the source position field) it would still be to no avail, since the value of $\delta$ would also be different, and the message could possibly run the risk of not being forwarded (by virtue of placing the receiver outside of the routing cone). Therefore, even if an attacker managed to act under multiple identities, there is no reward in doing so. □

**Theorem 4** *The direction-based algorithm is secure against blackhole and flooding.*

**Proof:**

Since the message-forwarding algorithm broadcasts the message, more than one path or channel is always created when a message is transmitted. Unless it was the only other node in the vicinity, a single adversary would not be able to interrupt communications by refusing to forward the data, as noted by Buttyán and Hubaux (2007). On the other hand, traffic restriction, which can be tuned to be sensitive to a large message frequency and/or cumulative payload, prevents nodes from launching flooding attacks and increases the message-forwarding resistance by lowering the probability of repeating a message originating from the general vicinity of the adversaries. □

# Chapter 6

# Conclusions and Future Work

## 6.1 Achievements

In this thesis, we prove that it is feasible to have private communications in a VANET while still ensuring message delivery by defining two algorithms that successfully deliver messages in networks. Both algorithms achieve this by encrypting messages and geocasting them across networks.

To better satisfy the communication requirements, we show a probabilistic and a deterministic routing method (direction-based model and transmission-coverage model, respectively). Throughout the analysis, we see that in the direction-based approach, the angle of spreading has a crucial role in communications, affecting the amount of messaging and the delivery success rate, while network density is of no consequence. On the other hand, the transmission-coverage approach is affected by network density yet while the angle of spreading modifies the latency of a message, it does not affect message delivery or network messaging overhead.

In general, the direction-based protocol has the benefit of controlling flooding and blackhole attacks, not being affected by node density and providing a degree of con-

trol over the levels of spreading of a message in a network. On the other hand, the transmission-coverage protocol generates a much smaller message overhead along with a higher level of reliability and has a more predictable routing protocol.

## 6.2 Future Work

### 6.2.1 Send Messages in Non-Linear Paths

One of the drawbacks of both algorithms is that they have a lower rate of reliability when sending messages around a corner. A possible solution to this problem is to employ a method similar to mixes. The message is wrapped in layers of headers, with each layer having a different destination and each destination being located in a straight line from the preceding source. The message is then forwarded from area to area. Every time the message arrives to an area, a layer is removed, and it is sent to the next area until it arrives at its desired final destination. This design is elegant in that it further improves on the privacy of the parties involved in the communication scenario. Since the message changes every time a layer is removed, it is very difficult to associate a source with a message.

### 6.2.2 Reduce Duplicated Messages

Regardless of the algorithm, the number of duplicated messages created is significant. More work on the subject could produce a solution that reduces this overhead on both algorithms. The direction-based algorithm, for one, assigns weights $w_i$ to messaging factors $k_i$ to determine the network traffic generated by an area. We did not test the impact different combinations of these variables would have on network transmission, but it is possible to produce values that optimize network load.

Both algorithms would also reduce overhead by using a directional antenna instead of an omnidirectional one to send messages towards a specific region. For example, in the direction-based model, nodes that are outside the angle of spread $\theta$ will never participate in routing the message, so any messages sent in their direction is a waste of resources. By sending messages only to the area defined by $\theta$, nodes outside the angle would not receive the unnecessary message. On the other hand, in the transmission-coverage model, forwarding nodes only need to send a message towards the uncovered area, since nodes iside the covered area have already received the message. If the message forwarding is constrained to the uncovered area, the number of duplicated messages is reduced. However, an optimization should be introduced to either algorithm only after it can be proven that it does not affect privacy.

### 6.2.3 Access Control

One logical improvement for both algorithms would be the addition of access control. Access control deals with allowing nodes with a set of properties to be able to read and/or write messages. By extension, access control also forbids nodes that do not possess those properties (or rights) from sending or reading a particular message. By itself, geocasting is a special case of access control, in which we grant read access to nodes in a particular region.

The general purpose of access control is twofold: it further restrains the number of nodes that can read a message and allows for innovative applications. An example of this is paying toll booths from a distance to avoid slowing down. A vehicle would send a message ahead to the line of toll booth with an encrypted message that contains its payment information. While other vehicles passing the toll booths can forward the message, only the pay booths can decrypt it.

Moreover, the network can be subdivided so that it can offer multi-tiered service. Essentially, groups with special privileges – such as police officers and ambulance dispatchers – can create messages that quickly identify and distinguish them from the rest to obtain preferential treatment in the messaging process. Another example is to create fleet-specific channels, so that fleet vehicles such as taxicabs can locate and communicate with other vehicles in the group seamlessly. These and other exciting applications are more appealing when privacy is ensured in the communications, which our algorithms ensure.

# References

Mahmoud Abuelela, Stephan Olariu, and Ivan Stojmenovic. OPERA: Opportunistic Packet Relaying in disconnected Vehicular Ad Hoc Networks. In *Proceedings of the IEEE 5th International Conference on Mobile Ad Hoc and Sensor Systems*, MASS, pages 285–294, Atlanta, GA, USA, October 2008. IEEE.

Carlisle Adams and Steve Lloyd. *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley Longman Publishing Company, Inc., Boston, MA, USA, 2nd edition, 2002. ISBN 978-0-672-32391-1.

Tiranuch Anantvalee and Jie Wu. A Survey on Intrusion Detection in Mobile Ad Hoc Networks. In Yang Xiao, Xuemin Sherman Shen, and Ding-Zhu Du, editors, *Wireless Network Security*, Signals and Communication Technology, chapter 7, pages 159–180. Springer US, 2007. ISBN 978-0-387-33112-6.

Boto Bako, Frank Kargl, Elmar Schoch, and Michael Weber. Advanced Adaptive Gossiping Using 2-hop Neighborhood Information. In *Proceedings of the Global Telecommunications Conference*, GLOBECOM, pages 1–6, New Orleans, LA, USA, December 2008. IEEE.

Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A Distance Routing Effect Algorithm for Mobility (DREAM). In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MOBICOM, pages 76–84, Dallas, TX, USA, October 1998. ACM.

Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenovic, editors. *Mobile Ad Hoc Networking*. Wiley Online Library, June 2004. ISBN 978-0-471-65689-0.

Alastair R. Beresford and Frank Stajano. Mix Zones: User Privacy in Location-Aware Services. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PerCom, pages 127–131, Orlando, FL, USA, March 2004. IEEE.

Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, Proceedings on the 24th Annual International Cryptology Conference*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55, Santa Barbara, CA, USA, August 2004a. Springer. ISBN 978-3-540-22668-0.

Dan Boneh, Ben Lynn, and Hovav Shacham. Short Signatures from the Weil Pairing. *Journal of Cryptology*, 17(4):297–319, January 2004b.

Levente Buttyán and Jean-Pierre Hubaux. *Security and Cooperation in Wireless Networks*. Cambridge University Press, 2007. ISBN 978-0-521-87371-0.

Levente Buttyán, Tamás Holczer, and István Vajda. On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs. In Stajano et al. (2007), pages 129–141. ISBN 978-3-540-73274-7.

Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and Robust Pseudonymous Authentication in VANET. In Wieland Holfelder, Paolo Santi, Yih-Chun Hu, and Jean-Pierre Hubaux, editors, *Proceedings of the 4th International Workshop on Vehicular Ad Hoc Networks*, pages 19–28, Montréal, Québec, Canada, September 2007. ACM. ISBN 978-1-59593-739-1.

*Canadian Motor Vehicle Traffic Collision Statistics, 2009 (Transport Canada).* Canadian Council of Motor Transport Administrators, May 2011. Available at http://www.tc.gc.ca/media/documents/roadsafety/tp3322-2009_eng.pdf.

Arnaud Casteigts, Amiya Nayak, and Ivan Stojmenovic. Communication Protocols for Vehicular Ad Hoc Networks. *Wireless Communications and Mobile Computing*, 11(5): 567–582, May 2011.

David Chaum and Eugène van Heyst. Group Signatures. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Proceedings of the Workshop on the Theory and Application of of Cryptographic Techniques*, volume 547 of *EUROCRYPT*, pages 257–265, Brighton, UK, April 1991. Springer. ISBN 978-3-540-54620-7.

Giacomo de Meulenaer, François Gosset, François-Xavier Standaert, and Olivier Pereira. On the Energy Cost of Communication and Cryptography in Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, WiMob, pages 580–585, Avignon, France, October 2008. IEEE.

Augusto J. Devegili, Michael Scott, and Ricardo Dahab. Implementing Cryptographic Pairings over Barreto-Naehrig Curves. In Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto, editors, *Proceedings of the First International Conference on Pairing-Based Cryptography*, volume 4575 of *Lecture Notes in Computer Science*, pages 197–207. Springer Berlin Heidelberg, Tokyo, Japan, July 2007. ISBN 978-3-540-73488-8.

John R. Douceur. The Sybil Attack. In Peter Druschel, M. Frans Kaashoek, and Antony I. T. Rowstron, editors, *Proceedings of the 1st International Workshop on Peer-to-Peer Systems, IPTPS, Revised Papers*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260, Cambridge, MA, USA, March 2002. Springer. ISBN 978-3-540-44179-3.

Karim El Defrawy and Gene Tsudik. PRISM: Privacy-friendly Routing in Suspicious MANETs (and VANETs). In *Proceedings of the 16th Annual IEEE International Conference on Network Protocols*, ICNP, pages 258 –267, Orlando, FL, USA, October 2008. IEEE.

Laura Marie Feeney. Energy-Efficient Communication in Ad Hoc Wireless Networks. In Basagni et al. (2004), chapter 11, pages 301–327. ISBN 978-0-471-65689-0.

Laura Marie Feeney and Martin Nilsson. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. In *Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, pages 1548–1557, Anchorage, AK, USA, April 2001. IEEE.

Andreas Festag, Panos Papadimitratos, and Tessa Tielert. Design and Performance of Secure Geocast for Vehicular Communication. *IEEE Transactions on Vehicular Technology*, 59(5):2456–2471, June 2010.

Julien Freudiger, Maxim Raya, Márk Felegyhazi, Panos Papadimitratos, and Jean-Pierre Hubaux. Mix-Zones for Location Privacy in Vehicular Networks. In *Proceedings of the 1st International Workshop on Wireless Networking for Intelligent Transportation Systems*, WiN-ITS, Vancouver, British Columbia, August 2007. ICST.

Julien Freudiger, Mohammad Hossein Manshaei, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. On the Age of Pseudonyms in Mobile Ad Hoc Networks. In *Proceedings of the 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, pages 1577–1585, San Diego, CA, USA, March 2010. IEEE. ISBN 978-1-4244-5838-7.

Antoine Gallais, Jean Carle, David Simplot-Ryl, and Ivan Stojmenovic. Localized Sensor Area Coverage with Low Communication Overhead. *IEEE Transactions on Mobile Computing*, 7(5):661–672, May 2008.

Sameh Gobriel, Daniel Mossé, and Rami Melhem. Mitigating the FloodingWaves Problem in Energy-Efficient Routing for MANETs. In *Proceedings on the 26th IEEE International Conference on Distributed Computing Systems*, ICDCS, page 47, Lisboa, Portugal, July 2006. IEEE. ISBN 978-0-7695-2540-2.

Michael M. Groat, Wenbo He, and Stephanie Forrest. KIPDA: k-Indistinguishable Privacy-preserving Data Aggregation in Wireless Sensor Networks. In *Proceedings of the 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies*, INFOCOM, pages 2024–2032, Shanghai, China, April 2011. IEEE.

Charles Harsch, Andreas Festag, and Panos Papadimitratos. Secure Position-Based Routing for VANETs. In *Proceedings of the 66th IEEE Vehicular Technology Conference*, VTC Fall, pages 26–30, Baltimore, MD, USA, September 2007. IEEE.

Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A Survey of Energy Efficient Network Protocols for Wireless Networks. *Wireless Networks*, 7(4):343–358, September 2001. ISSN 1022-0038.

Marc Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In *Proceedings of the 4th International Conference on Ubiquitous Computing, UbiComp*, volume 2498 of *Lecture Notes in Computer Science*, pages 315–320, Göteborg, Sweden, October 2002. Springer.

Fan Li and Yu Wang. Routing in Vehicular Ad Hoc Networks: A Survey. *Vehicular Technology Magazine*, 2(2):12–22, June 2007. ISSN 1556-6072.

Ben Lynn. *On the Implementation of Pairing-Based Cryptosystems*. PhD thesis, Stanford University, June 2007. Available at http://crypto.stanford.edu/pbc/.

Pietro Michiardi and Refik Molva. Ad Hoc Networks Security. In Basagni et al. (2004), chapter 12, pages 329–354. ISBN 978-0-471-65689-0.

Akira Mizumoto, Hirozumi Yamaguchi, and Kenichi Taniguchi. Cost-Conscious Geographic Multicast on MANET. In *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON, pages 44–53, Santa Clara, CA, USA, October 2004. IEEE.

*Early Estimate of Motor Vehicle Traffic Fatalities in 2011 (U.S. Department of Transportation)*. National Highway Traffic Safety Administration, May 2012. Available at http://www-nrd.nhtsa.dot.gov/Pubs/811604.pdf.

Panos Papadimitratos and Zygmunt J. Haas. Secure Routing for Mobile Ad Hoc Networks. In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, CNDS, San Antonio, TX, USA, 2002. SCS.

Panos Papadimitratos, Levente Buttyan, Tamás Holczer, Elmar Schoch, Julien Freudiger, Maxim Raya, Zhendong Ma, Frank Kargl, Antonio Kung, and Jean-Pierre Hubaux. Secure Vehicular Communication Systems: Design and Architecture. *Communications Magazine, IEEE*, 46(11):100 –109, November 2008. ISSN 0163-6804.

Bryan Parno and Adrian Perrig. Challenges in Security Vehicular Networks. In *Proceedings of the 4th ACM Workshop on Hot Topics in Networks*, HotNets-IV, College Park, MA, USA, 2005. ACM.

Andreas Pfitzmann and Marit Köhntopp. Anonymity, Unobservability, and Pseudonymity — A Proposal for Terminology. In Hannes Federrath, editor, *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9, Berkeley, CA, USA, July 2001. Springer. ISBN 978-3-540-41724-8. A regularly updated version of this paper is maintained at http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.

Sanjay Ramaswamy, Huirong Fu, Manohar Sreekantaradhya, J. Dixon, and Kendall Nygard. Prevention of Cooperative Black Hole Attack in Wireless Ad Hoc Networks. In *Proceedings of the International Conference on Wireless Networks*, ICWN, Las Vegas, NV, USA, June 2003. CSREA.

Thomas A. Ranney, Elizabeth Mazzae, Riley Garrott, and Michael J. Goodman. NHTSA Driver Distraction Research: Past, Present and Future. In *Proceedings of the Internet Forum on Driver Distraction*. NHTSA, 2000.

Maxim Raya. *Data-Centric Trust in Ephemeral Networks*. PhD thesis, École Polytechnique Fédérale de Lausanne, Lausanne, June 2009.

Maxim Raya and Jean-Pierre Hubaux. The Security of Vehicular Ad Hoc Networks. In *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, SASN, pages 11–21, New York, NY, USA, 2005. ACM. ISBN 978-1-59593-227-3.

Maxim Raya and Jean-Pierre Hubaux. Securing Vehicular Ad Hoc Networks. *Journal of Computer Security*, 15(1):39–68, 2007.

Amit Sahai and Brent Waters. Fuzzy Identity-Based Encryption. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473, Aarhus, Denmark, May 2005. Springer. ISBN 978-3-540-25910-7.

Krishna Sampigethaya, Leping Huang, Mingyan Li, Radha Poovendran, Kanta Matsuura, and Kaoru Sezaki. CARAVAN: Providing Location Privacy for VANET. In *Proceedings of the Embedded Security in Cars Workshop*, ESCAR, Cologne, Germany, 2005.

Krishna Sampigethaya, Mingyan Li, Leping Huang, and Radha Poovendran. AMOEBA: Robust Location Privacy Scheme for VANET. *IEEE Journal on Selected Areas in Communications*, 25(8):1569–1589, 2007.

Elmar Schoch, Frank Kargl, Michael Weber, and Tim Leinmuller. Communication Patterns in VANETs. *Communications Magazine, IEEE*, 46(11):119–125, November 2008.

Elmar Schoch, Boto Bako, Stefan Dietzel, and Frank Kargl. Dependable and Secure Geocast in Vehicular Networks. In *Proceedings of the 7th ACM International Workshop on VehiculAr InterNETworking*, VANET, pages 61–68, Chicago, IL, USA, 2010. ACM.

David Simplot-Ryl, Ivan Stojmenović, and Jie Wu. Energy-Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks. In Ivan Stojmenovic, editor, *Handbook of Sensor Networks*, pages 343–380. Wiley Online Library, September 2005. ISBN 978-0-471-74414-6.

Frank Stajano, Catherine Meadows, Srdjan Capkun, and Tyler Moore, editors. *Proceedings of the 4th European Workshop on Security and Privacy in Ad-hoc and Sensor Networks, ESAS*, volume 4572 of *Lecture Notes in Computer Science*, Cambridge, UK, July 2007. Springer. ISBN 978-3-540-73274-7.

Ivan Stojmenovic, Anand P. Ruhil, and D. K. Lobiyal. Voronoi Diagram and Convex Hull Based Geocasting and Routing in Wireless Networks. *Wireless Communications and Mobile Computing*, 6(2):247–258, March 2006.

Ahren Studer, Elaine Shi, Fan Bai, and Adrian Perrig. Efficient Mechanisms to Provide Convoy TACKing Together Efficient Authentication Revocation, and Privacy in VANETs. In *Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON, pages 1–9, Rome, Italy, June 2009.

Di Tian and Nicolas D. Georganas. Location and Calculation-Free Node-Scheduling Schemes in Large Wireless Sensor Networks. *Ad Hoc Networks*, 2(1):65–85, January 2004.

Yu-Chee Tseng, Sze-Yao Ni, Yuh-Shyan Chen, and Jang-Ping Sheu. The Broadcast Storm Problem in a Mobile Ad Hoc Network. *Wireless Networks*, 8:153–167, March 2002. ISSN 1022-0038.

Andrew M. White, Austin R. Matthews, Kevin Z. Snow, and Fabian Monrose. Phonotactic Reconstruction of Encrypted VoIP Conversations: Hookt on Fon-iks. In *Proceedings on the 32nd IEEE Symposium on Security and Privacy*, S&P, pages 3–18, Berkeley, CA, USA, May 2011. IEEE.

Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-Vehicle Safety Messaging in DSRC. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET, pages 19–28, Philadelphia, PA, USA, October 2004. ACM.

Fan Ye, Gary Zhong, Jesse Cheng, Songwu Lu, and Lixia Zhang. PEAS: A Robust Energy Conserving Protocol for Long-Lived Sensor Networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ICDCS, pages 28–37, Providence, RI, USA, May 2003. IEEE.