Applying transmission-coverage algorithms for secure geocasting in VANETs

Antonio Prado*

SEECS,
University of Ottawa,
800 King Edward Ave., Ottawa ON K1N 6N5, Canada
Email: aprad046@uottawa.ca
*Corresponding author

Sushmita Ruj

Indian Statistical Institute,
Kolkata, 203 B.T. Road,
Kolkata 700108, India
Email: sush@isical.ac.in

Milos Stojmenovic

Singidunum University,
Danijelova 32, 11000,
Belgrade, Serbia
Email: mstojmenovic@singidunum.ac.rs

Amiya Nayak

SEECS, University of Ottawa,
800 King Edward Ave.,
Ottawa ON K1N 6N5, Canada
Email: nayak@uottawa.ca

Abstract: Existing geocasting algorithms for VANETs provide either high availability or security, but fail to achieve both together. Most of the privacy preserving algorithms for VANETs have low availability and involve high communication and computation overheads. The reliable protocols do not guarantee secrecy and privacy. We propose a secure, privacy-preserving geocasting algorithm for VANETs, which uses direction-based dissemination. Privacy and security are achieved using public key encryption and authentication and pseudonyms. To reduce communication overheads resulting from duplication of messages, we adapt a transmission-coverage algorithm used in mobile sensor networks, where nodes delay forwarding messages based on its uncovered transmission perimeter after neighbouring nodes have broadcast the message. Our analysis shows that our protocol achieves a high delivery rate, with reasonable computation and communication overheads.

Keywords: geocasting; privacy; coverage; vehicular ad-hoc network; VANET.

Reference to this paper should be made as follows: Prado, A., Ruj, S., Stojmenovic, M. and Nayak, A. (2018) 'Applying transmission-coverage algorithms for secure geocasting in VANETs', *Int. J. Computational Science and Engineering*, Vol. 16, No. 1, pp.17–26.

Biographical notes: Antonio Prado received his Master in Computer Science from the University of Ottawa. His work in academia as well as in the private sector revolves around computer networks and communication protocols. Currently, he works with high availability emergency response networks.

Sushmita Ruj received her BE in Computer Science from Bengal Engineering and Science University, Shibpur, India in 2004, and Master's and PhD in Computer Science from Indian Statistical Institute, India in 2006 and 2010, respectively. She was a Post-Doctoral Fellow at Lund University, Sweden and at University of Ottawa, Canada. She was an Assistant Professor at Indian Institute of Technology, IIT, Indore and is currently an Assistant Professor at Indian Statistical Institute, Kolkata. Her research interests are in security and privacy in mobile ad hoc networks, vehicular networks, smart grids, social networks, cloud computing, combinatorics and cryptography.

Milos Stojmenovic completed his PhD in Computer Science at the University of Ottawa, Canada in 2008. Currently, he is an Associate Professor at Singidunum University, Serbia. He was a Visiting Researcher at Japan's National Institute of Advanced Industrial Science and Technology in 2009. He has published over thirty articles in the fields of computer vision, image processing, and wireless networks. His work implements machine learning techniques such as AdaBoost and SVM classification which in turn use higher order autocorrelation features (HLAC) to perform image segmentation and classification.

Amiya Nayak received his BMath in Computer Science and Combinatorics and Optimisation from the University of Waterloo in 1981, and his PhD in Systems and Computer Engineering from Carleton University in 1991. He has over 17 years of industrial experience, working at CMC Electronics, Defence Research Establishment Ottawa and Nortel Networks. Currently, he is a Full Professor at the School of Electrical Engineering & Computer Science at the University of Ottawa. His main research interests are in the areas of fault tolerance, wireless sensor networks, and distributed systems, with over 200 publications in refereed journals and conference proceedings.

1 Introduction

Vehicular ad-hoc networks (VANETs) consist of vehicles, road side units (RSUs) and other infrastructure that can provide safe and smooth driving. VANETs provide information about road conditions and traffic information as well as location-based services for a comfortable experience on the road. Vehicles are equipped with GPS and sensors that sense and gather information and pass to other vehicles.

There are many algorithms for communications in a VANET (Schoch et al., 2008; Li and Wang, 2007). Geocasting (Casteigts et al., 2011) is a form of one-to-many communication. In geocasting-based protocols, messages are delivered to vehicles within a specific geographical region. We argue that both privacy and reliability are required attributes to have useful communications in a VANET. Private messages that never arrive waste network resources, while insecure messaging limits the applications of the network to general information only.

Currently, most VANET applications deliver non-private information, such as road conditions or traffic levels, where 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 useful if the information can be read by every vehicle in the network. Versatile communication algorithms must then allow confidential communications using encryption. However, encryption is not enough for privacy, since knowledge of messaging behaviour such as messaging frequency and payload sizes can still be used to compromise privacy (Raya and Hubaux, 2005, 2007). A scenario where anonymous geocast communication is very useful is warning preceding vehicles of bad road conditions (e.g., ice, accident, etc.). The sender does not know which specific vehicle might be interested in this information, but knows that nearby vehicles within a certain range might find this useful, so it geocasts to that area. Vehicles in the area do not need the identity of the benefactor to receive the information.

Current VANET geocasting algorithms are designed to ensure either privacy or reliability, but not both. Privacy-driven geocasting algorithms (El Defrawy and Tsudik, 2008; Festag et al., 2010) cannot ensure that the message will arrive to its destination. Conversely, algorithms which ensure message delivery (Basagni et al., 1998; Stojmenovic et al., 2006) have no privacy.

1.1 Our contribution

In this paper, we propose a reliable and privacy-preserving geocasting algorithm that uses transmission perimeter coverage algorithm technique used in wireless sensor networks for energy conservation and to avoid message duplication (Simplot-Ryl et al., 2005). Messages are encrypted using public key encryption techniques and signed to verify the authenticity. To obtain privacy, we use pseudonyms (Pfitzmann and Köhntopp, 2001) so that forwarding nodes only release the current location of the node. 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. The dissemination method is a variation of the transmission-area coverage used in connected dominating sets (CDS) algorithms (Simplot-Ryl et al., 2005). Using analysis and simulations, we show that our protocol is secure, ensures authentication of nodes, preserves privacy and communicates reliably.

1.2 Organisation

The paper is organised as follows: Section 2 reviews previous work related to the subject and the assumptions made for the attacker model. Section 3 will then describe in

detail the protocol we are proposing, which is then evaluated in Section 4. We conclude in Section 5.

2 Related work

There are many routing and geocasting algorithms for MANETs such as VD-GREEDY and CH-MFR (Stojmenovic et al., 2006). However, they do not preserve the privacy of the nodes, nor are they suitable for ephemeral networks like VANETs. Their routing processes optimally forward the message to the neighbour that is closest to the destination (i.e., progresses towards the destination). The VD-GREEDY method determines these neighbours using Voronoi diagram of neighbours that are near the destination area, while CH-MFR uses the convex hull of neighbouring nodes. The selection of forwarding nodes is optimal with respect to power consumption, which is a concern in MANETs. However, power consumption is not a concern in VANETs, where nodes are connected to a generous power supply. The main privacy concern with these algorithms is that they require a level of topology knowledge – the position of neighbouring node – to effectively select a node to forward to. This information could be used by a malicious node to deduce the identity of a specific node.

A major concern for geocasting is to find an efficient set of nodes to transmit messages, since message flooding wastes bandwidth. To address this, some MANETs use CDS routing algorithms. A CDS is 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) to provide redundancy while minimising energy expenditure. Once again, the main privacy concern with this model is that it also requires a high level of topology knowledge. Though these techniques transmit messages efficiently, these cannot be used in VANETs because they guarantee neither security nor privacy.

Secure VANETs need confidentiality, message authentication and integrity (Papadimitratos et al., 2008). Pseudonyms are used to achieve privacy and authentication (Sampigethaya et al., 2007). Pseudonyms are aliases that a node uses to communicate, while keeping its real identity hidden, which allows it to retain its privacy. 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 where an adversary could place a vehicle in different locations at different times and observe a connection or pattern. However, 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 to its destination, making them private but not reliable.

A protocol which ensures privacy is PRISM (El Defrawy and Tsudik, 2008) which encrypts data using group keys and sends messages to a geographical area rather than a specific node. 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 reduce 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 not useful any longer.

Another element of private communications is unlinkability, the situation where adversary cannot sufficiently distinguish whether two nodes (e.g., sender and receiver) are related (Pfitzmann and Köhntopp, 2001). The degree of unlinkability between two nodes i, j in a network of n nodes is then the entropy of all possible links between i, j (Steinbrecher and Kopsell, 2003).

2.1 Attacker model

An attacker can only directly communicate with the nearest neighbour. No single attacker has global knowledge of network topology. Multiple attackers are non-colluding. 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. An attacker can be either active or passive. A passive attacker will only be able to read the message headers, and store them for forensic analysis. An active attacker can listen to the communications and read the message headers, modify them, and selectively inject them to the network, to interfere with communications. Attackers could also try to flood the system with replay attacks, by constantly repeating an invalid message.

2.2 Mathematical techniques and signature schemes used

Throughout this paper, we will use the notation defined in Table 1. 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 $e : G \times G \rightarrow G_T$. This efficiently computable, non-degenerate ($e(g, g) = 1$) map also satisfies the property $e(g^a, g^b) = e(g, g)^{ab}$, $a, b \in \mathbb{Z}_q$. We use BLS signatures (Boneh et al., 2004) to authenticate the message validity in three steps:

1 Set-up: The key distribution centre (KDC) chooses a group G of order q and each user chooses a secret key $x \in \mathbb{Z}_q$. Let f be a generator of G and $H(.)$ a hash function, such that $H: \{0, 1\}^* \rightarrow G$. The public key of the user is then $X = f^x$.

2 Sign: A node sending message m_i creates a signature $\sigma = H(m_i)^x \in G$.

3 Verify: A signature σ is verified by checking

$$\begin{aligned} e\left(H\left(m_{\rightarrow}\right), X\right) &= e\left(H\left(m_i\right), f^x\right) \\ &= e\left(H\left(m_i\right)^x, f\right) \\ &= e(\sigma, f). \end{aligned} \tag{1}$$

Table 1 Notation used in the paper

Notation	*Meaning*
N	Number of nodes in the network
H_i	i^{th} node
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 $m_i \in M$

3 The protocol

We consider a network of n vehicles, RSUs and certification authority. Each vehicle n_i receives multiple pseudonyms and corresponding public/secret keys from the CA. When a vehicle sends a message, it sets the routing and broadcasting headers. The initial sender defines the *DST_AREA* and a point C inside the area, as well as the arc of interest θ used for routing (to be discussed later). The values in 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+i} 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. 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.

Each message has a message header that contains the routing information, as seen in Figure 1. After all the headers are generated, the message is encrypted and signed. The data packet is then broadcast.

Figure 1 Message packet fields

DST_AREA	θ	Timestamp
Expiration Time	TTL	Sender Position
Protected Data		

Upon receipt of a message, a node automatically ignores invalid messages. For valid messages, instead of deciding whether to discard or forward immediately, the node checks the transmission perimeter 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 and increases or resets the amount of time it will wait to avoid redundant broadcasting. The duration of the transmission delay is also affected by the location and number of requests to forward the message in question. 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 neighbours as possible, while the main objective of transmission outside the area is to reliably deliver the message to the destination.

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 messages. Nodes outside the destination area also accelerate the rate of retransmission when nearing the destination by reducing the lengths of the delays.

3.1 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:

1 the transmission perimeter of the vehicle that has not received the message

2 the distance between the vehicle and the centre of the destination area

3 a random value used to resolve ties between vehicle retransmissions.

3.1.1 Distance to centre

When a node receives a copy of the message, it determines how close it is to the centre 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 centre C, transmission continues until the *DST_AREA* is

fully covered. If vehicle A is closer than vehicle B to the destination centre C, it will retransmit the message first. 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, $t_A \propto |\overrightarrow{CA}|$.

3.1.2 Transmission perimeter

Every vehicle (node) has a transmission range that covers a circular perimeter of radius r around it. When a node transmits a message, it can assume that every node inside the transmission range (i.e., less than the distance r) has received a copy of the message. Likewise, when a node receives a message from another node, it knows that all the nodes in the transmission range of that node – the receiving node included – have received the message. The receiving node can also calculate the perimeter where the transmission of both the sender node and its own overlap, and is aware that all the nodes inside it have received a copy of the message. We call this overlap the *covered perimeter*. If the receiving vehicle were to broadcast the message, it would only be useful to those nodes that are outside of the overlap (i.e., the uncovered perimeter).

To avoid sending a duplicated message, when a copy of the message is received, the node calculates the covered perimeter, A_C and updates the transmission delay time t. Since a larger A_C translates to a smaller uncovered perimeter, the larger A_C is, the less useful a rebroadcast is, so it can be further delayed. For that reason, a larger A_C implies a longer t, such that $t \propto A_C$.

If the perimeter of the node is fully covered by other nodes, it decides that it is not necessary to forward the message. 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 details on how coverage is calculated, see Section 3.3.

3.1.3 Random value

To reduce the probability of a tie – e.g., when two vehicles are within the same zone and have the same coverage – a random value R is used. The benefit of minimising ties is that if two vehicles are within range of each other and when one receives a duplicate message from the other, it simply increases the delay for the message to be retransmitted. It is possible that the vehicle can forgo transmitting the message altogether, thus minimising duplicates. Combining these values, we get $t = |CA|A_CR$.

3.2 Outside the geocast area

Outside 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 they move the message closer to the destination area. To determine if 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 θ 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.

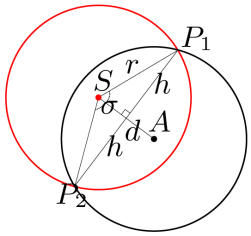
3.3 Transmission perimeter coverage

Upon receipt of a message M, the vehicle determines the perimeter 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 perpendicular to $\overrightarrow{SA}$ and passes over the points P_1, P_2.

Once the points are obtained, the source calculates the angle σ formed by P_1AP_2 and the circular segment perimeter C_a, which is the perimeter where the two circles overlap. See Figure 2 for a visual example of this process. The vehicle determines P_1, P_2 and C_a as follows:

$$V_\Sigma = (A+S)/2 \qquad P_1(x) = V_\Sigma(x) + h/d * V_\Delta(y)$$
$$V_\Delta = A - S \qquad P_2(x) = V_\Sigma(x) - h/d * V_\Delta(y)$$
$$d = |V_\Delta| \qquad P_1(y) = V_\Sigma(y) - h/d * V_\Delta(x)$$
$$h^2 = r^2 - (d/2)^2 \qquad P_2(y) = V_\Sigma(y) + h/d * V_\Delta(x)$$
$$\cos\sigma = \frac{|\overrightarrow{P_1A}|^2 + |\overrightarrow{P_2A}|^2 - |\overrightarrow{P_1P_2}|^2}{2|\overrightarrow{P_1A}||\overrightarrow{P_2A}|}$$
$$C_a = r\sigma, \quad C = \frac{C_a}{2\pi r} = \frac{\sigma}{2\pi}$$

Figure 2 Transmission coverage overlap (see online version for colours)



Note: The intersection between S and A determines coverage.

Given this information, the receiving vehicle can determine the perimeter in its transmission range that has also received the message and therefore does not need to receive a duplicate of the message. The complement of this is the perimeter that has not yet received the message. As the uncovered perimeter is reduced, so is the probability that there is a node in the transmission range that needs to receive the message. Therefore, nodes that have the smallest uncovered perimeter also have the least urgency to retransmit the message, while nodes that have a smaller covered perimeter 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 perimeter covered $C = C_a/2\pi r$ and updates the transmission delay accordingly.

Once the entire transmission perimeter is covered (i.e., $C_a = 2\pi r$), 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 creates a duplicate. For this reason, the forwarding node decides not to forward the message.

We treat the perimeter as a line segment of length of 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 different transmissions cover the perimeters l_1, l_2, it is modelled 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 an overlap. Namely, perimeter l_2 overlaps with l_1 when $l_1 \cap l_2 \neq \emptyset$, which is defined as follows:

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

We have:

$$L_a = \bigcup_i l_i$$

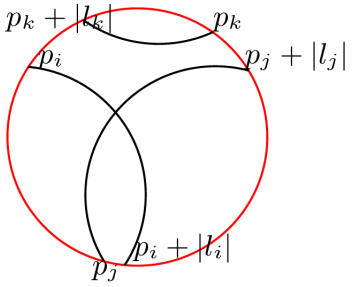
where $\forall i, ji \neq j$

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

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

This way, l_i and l_j are merged when they overlap, otherwise they are kept as separate parameters. Figure 3 shows how two perimeters might intersect.

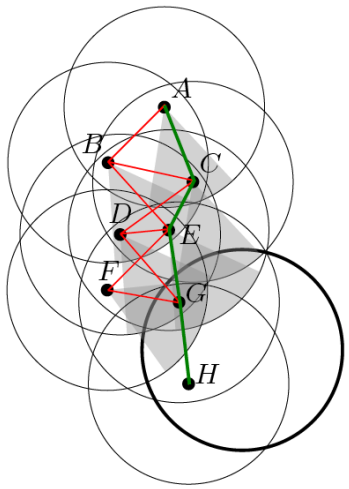
Figure 3 Overlapping perimeters (see online version for colours)



An example of the behaviour of the algorithm follows. Figure 4 shows the routing process.

There are seven cars on this part of the highway when A wants to send a message to an area behind it. Both H and G are on the destination area, so they will be able to decode the message. A sets the angle of interest to $\pi/6$ and broadcasts the message which arrives at C and B. After the broadcast, A ignores any further copies of the message that it receives. Since C is closer to the destination, it will have a slightly smaller delay than B. Therefore, C will broadcast next, and its message reaches B, D and E. Car B still has a small part of its arc uncovered, so it will increase its broadcast delay. Car E will broadcast next since it is the closest to the destination, and its broadcast is received by D, B, F and G. When D and B receive the message from E and update their covered perimeters, they notice that the arc of interest is covered, so they do not broadcast the message. Since G is inside the destination area, it is very likely that it will broadcast before F, and its message will reach F and H. F does not need to broadcast the message once it receives a copy from G, while H will repeat the message when its delay is over.

Figure 4 Routing example (see online version for colours)



In the above scenario, if the random delay at E caused it to broadcast after D, the message would still arrive at the destination area, routed through D instead of E.

3.4 Encryption mechanism

To transmit securely, we encrypt the message such that only intended recipients belonging to a particular geographic area can decode it. Let *PK*/*SK* be the public/private key pair for a pseudonym and cert be the corresponding public key certificate. A KDC is appointed to be responsible of managing the secret credentials of vehicles. The KDC chooses $a_1, a_2, ..., a_v \in \mathbb{Z}_q$ randomly, for each zone $A_1, A_2, ..., A_v$ and a secret $y \in \mathbb{Z}_q$. These are only known by the KDC and are regularly refreshed. The KDC securely chooses a group G and G_T of order q, a generator $g \in G$ and calculates $g^{a_1}, g^{a_2}, ..., g^{a_v}$. $Y = e(g, g)^y$ is calculated. The KDC provides the RSU in the geographic location Ar with the secret key g^{y/a_r}.

The secret parameters of the protocol $SPM = \langle a_1, a_2, ..., a_v, y \rangle$ are determined, and the public parameters $PPM = \langle g^{a_1}, g^{a_2}, ..., g^{a_v}, \Psi \rangle$ are sent by the nearest RSU when a vehicle enters a geographical location. By presenting its pseudonymous ID to an RSU in region A_r, the vehicle receives a geographic group secret key g^{y/a_r} encrypted by the vehicle's public key *PK*. This deters vehicles with invalid credentials from reading the key g^{y/a_r} and thus disables them from decrypting information that they are not supposed to receive. Encryption and decryption proceed as follows:

1 Encryption: a vehicle wishing to send a message m to a specific geographic location A_r:

- chooses $s \in \mathbb{Z}_q$ randomly
- calculates Y^s
- calculates $C_0 = g^a r^s$ from g^{a_r}
- sends ciphertext $C = \langle mY^s, C_0 \rangle$.

2 Decryption: when a message arrives, a node (vehicle) calculates the following:

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

From mY^s and Y^s, the message m can be computed. The problem with this approach is that any intermediary node can change the message and pretend to be a valid sender. Thus, we need to authenticate the source node who has sent the message.

3.5 Authentication

The encrypted message C together with the *DST_AREA*, timestamp, expiration time and θ message headers is signed by the sender. This ensures that intermediate nodes are not able to modify any of these parameters. The concatenated message

$$m = \langle C \,||\, DST\_AREA \,||\, Timestamp \,||\, Expiration\_time \,||\, \theta \rangle$$

is signed using the protocol in 2.2, while the signature σ is calculated using $\sigma = H(m_i)^x \in G$, where x is the secret key chosen for the sender. The sender then broadcasts $sig = (C, \sigma, cert)$ along with the header. The protected message is then padded with zeros so that length of all messages are the same.

When a node receives the information *sig*, it verifies the certificate *cert* (Raya and Hubaux, 2005). Signature σ is then checked using equation (1). If the message is unmodified, the verification proceeds smoothly.

4 Performance evaluation

To gather empirical data, we used the *Ns*-3 network simulator (Henderson et al., 2006) with a VANET network infrastructure. We ran the simulations in a 4×4 km area for a period of 60 s of network operation. We ran scenarios to see how the algorithm behaves at different levels of network density (500–1,500 nodes), and we also saw the effect that θ has over the communications.

4.1 Experimental results

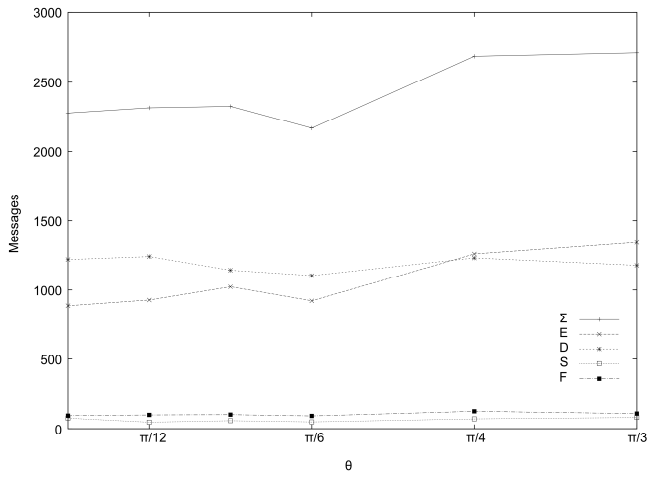
Experimentally, we found that the transmission radius of the vehicles in the network was around 120 m. While more powerful or accurate devices could be conceived, we believe that this transmission radius is effective for the messaging operations at hand.

This algorithm is influenced by the number of nodes in the network. 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 range, increasing their probability of being fully covered and deciding not to transmit. Therefore, node density subtly shapes network traffic.

When routing, a larger value of θ translates to a larger uncovered transmission perimeter, which decreases the initial transmission delay and would ostensibly increase the number of forwarding nodes and/or routing paths. In other words, if $\theta = \pi$, 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 perimeter is smaller than normal. Figure 5 shows that, indeed, the number of messages used to calculate the covered perimeter increases, yet the number of messages forwarded does not seem to be affected by different values of θ.

Figure 5 Relationship between θ and routing overhead



4.1.1 Computational costs

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

During authentication, signing involves one hash computation and exponentiation, which requires 0.6 ms. Verification requires two pairing operations to calculate $e(H(C), X)$ and $e(\sigma, g)$, which require 9 ms.

4.1.2 Message size

To ensure integrity of message 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.

4.1.3 Node density and message overhead

Table 2 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 realises 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 perimeter coverage (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 Σ, strongly supports the theory that not all nodes need to forward the message upon receipt.

Total network traffic (Σ) 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.

4.1.4 Angle of coverage and message overhead

In Figure 5, we can see the effect θ has on the messaging overhead. As expected, a smaller value of θ translates to less coverage perimeter for the routing nodes and results in a smaller number of messages used for calculating the transmission perimeter (E), while at the same time increases the number of messages discarded upon receipt because the transmission perimeter 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 has already forwarded. The number of messages being discarded for this reason ($D - D_{full}$) grows as θ increases. Interestingly, the outcome of this is that the network load (Σ) does not seem to be affected by the value of θ.

Furthermore, the number of forwarded messages F does not seem to be affected by the value of θ, which goes against the original expectations. For these reasons, it is logical to conclude that the value of θ does not ultimately affect network traffic.

Table 3 shows the number of distinct messages that arrived in the *DST_AREA*, while Table 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 a *DST_AREA*. All the senders use the same parameters (*DST_AREA*, θ and initial sending time) and are all randomly placed on the network.

In Table 3, we show the reliability of the algorithm as well as the impact θ has on this property. We see that the larger θ is, the less reliable the algorithm becomes. We know that the only time θ affects the behaviour of the protocol is during the routing phase. From Figure 5, 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. Each item in the table represents the number of distinct messages that arrive in the destination area, with respect to the initiators and θ. For example, when $\theta = \pi/6$, and 15 senders create a message, 11 of 15 messages arrive at a node inside the *DST_AREA*, and the message is forwarded 12 times.

Table 2 Effect of node density on messaging overhead

$I = 1$, $\theta = \pi/6$							
N	S	F	D	D_{full}	E	Σ	$\frac{D+E}{\Sigma}$
500	49	92	1,104	945	922	2,167	0.935
750	65	117	2,621	2,400	1,811	4,614	0.961
1,000	143	118	3,398	3,164	2,538	6,197	0.958
1,250	154	110	4,357	4,113	3,021	7,642	0.966
1,500	212	97	4,890	4,711	3,332	8,531	0.964

Notes: Legend:
S: in *DST_AREA*. E: messages that affect coverage.
F: forwarded. D_{full}: discard by full coverage.
D: total discarded. $\Sigma = S + F + D + E$

Table 3 Distinct messages inside *DST_AREA*(S)

	S	I					
		5	*10*	*15*	*20*	*25*	*50*
θ	$\pi/12$	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

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 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.

Table 4 Distinct messages outside *DST_AREA*(F)

	F	I					
		5	*10*	*15*	*20*	*25*	*50*
θ	$\pi/12$	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

4.2 Security of our protocol

We will show that our protocol is secure and privacy preserving. We will first show that an insider attacker cannot read a message that it is not supposed to read. It cannot either pretend to be the sender and change the information.

Theorem 1: Our protocol is secure. A node cannot decrypt a message which 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 hard to calculate a given g^a (discrete logarithm problem). So, a vehicle cannot calculate a_1, a_2, ..., a_v or y given $g^{a_1}, g^{a_2}, ..., g^{a_v}$, 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 cannot be calculated. A node which has left region A_r and still has the secret key g^{y/a_r} can decrypt all messages to A_r, but due to key refreshment, the values of y and a_r change, so the node is unable to calculate any messages afterwards.

A vehicle which receives the ciphertext cannot change it and pretend to be the source. This is guaranteed by our signature scheme. The ciphertext is signed using signature σ 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 a 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 cannot also sign a randomly generated message and send it with a new certificate because a valid certificate from the trusted authority should exist. Thus, an adversary cannot change the content of the message without being detected. □

Theorem 2: Our protocol preserves privacy.

Proof: We prove this by showing that the communications are pseudonymous and unlinkable. Since only the CA knows the unique ID of a vehicle, with a sufficiently large amount of issued pseudonyms 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 every vehicle is the current location at the time of forwarding. To prove unlinkability, we show that for all possible observations an attacker can make, it cannot

determine a relationship between a sender node S and receiver node *C*.

Adversaries cannot identify a vehicle by the message forwarding pattern it uses since every node makes a forwarding decision using a random delay. If we let *T* be the set of nodes used in the communication paths over which a sender node transmits a message to the geocast region, we can determine the information $I(T; O)$ an attacker has over *C*. We have:

$$I(T;\ O) = \log \frac{P(O \mid T)}{P(O)} = \log \frac{1/n}{1/p} = \log \frac{p}{n}$$

Moreover, the message size *a* does not change during transmission, so the correlation between *O* and *a* is $I(a; O) = 0$. Since message headers do not contain any other unique information (e.g., vehicle id, destination, etc.), there is no secondary parameter that can be used to link a message with its forwarding node. Furthermore, the destination is also not a specific node, but a geographical location, which prevents the adversary from detecting a unique link between endpoints. Therefore, the communication channel is unlinkable and pseudonymous. □

5 Conclusions and future work

We propose a secure and privacy-preserving geocasting protocol using transmission coverage forwarding algorithm. Our algorithm is reliability and generates fewer duplicate messages. More work on the subject could produce a solution that reduces this overhead further. Using directional antennae to send messages towards a specific region would limit the number of nodes that receive a message. This and other exciting applications are more appealing when privacy is ensured in the communications, which our algorithm provides.

References

Basagni, S., Chlamtac, I., Syrotiuk, V.R. and Woodward, B.A. (1998) 'A distance routing effect algorithm for mobility (DREAM)', in *ACM MOBICOM*, pp.76–84.

Boneh, D., Lynn, B. and Shacham, H. (2004) 'Short signatures from the Weil pairing', *Journal of Cryptology*, Vol. 17, No. 4, pp.297–319.

Casteigts, A., Nayak, A. and Stojmenovic, I. (2011) 'communication protocols for vehicular ad hoc networks', *Wireless Communications and Mobile Computing*, Vol. 11, No. 5, pp.567–582.

Devegili, A.J., Scott, M. and Dahab, R. (2007) 'Implementing cryptographic pairings over Barreto-Naehrig curves', in *Pairing*, Vol. 4575, pp.197–207.

El Defrawy, K. and Tsudik, G. (2008) 'PRISM: privacy-friendly routing in suspicious MANETs (and VANETs)', in *IEEE International Conference on Network Protocols*, pp.258–267.

Festag, A., Papadimitratos, P. and Tielert, T. (2010) 'Design and performance of secure geocast for vehicular communication', *IEEE Transactions on Vehicular Technology*, Vol. 59, No. 5, pp.2456–2471.

Henderson, T.R., Roy, S., Floyd, S. and Riley, G.F. (2006) 'Ns-3 project goals', in *Proceeding of the 2006 Workshop on Ns-2: The IP Network Simulator*.

Li, F. and Wang, Y. (2007) 'Routing in vehicular ad hoc networks: a survey', *IEEE Vehicular Technology Magazine*, Vol. 2, No. 2, pp.12–22.

Papadimitratos, P., Buttyan, L., Holczer, T., Schoch, E., Freudiger, J., Raya, M., Ma, Z., Kargl, F., Kung, A. and Hubaux, J-P. (2008) 'Secure vehicular communication systems: design and architecture', *IEEE Communications Magazine*, Vol. 46, No. 11, pp.100–109.

Pfitzmann, A. and Köhntopp, M. (2001) 'Anonymity, unobservability, and pseudonymity – a proposal for terminology', in *Workshop on Design Issues in Anonymity and Unobservability*, pp.1–9.

Raya, M. and Hubaux, J-P. (2005) 'The security of vehicular ad hoc networks', in V. Atluri, P. Ning and W. Du (Eds.): *Workshop on Security of Ad Hoc and Sensor Networks*, pp.11–21.

Raya, M. and Hubaux, J-P. (2007) 'Securing vehicular ad hoc networks', *Journal of Computer Security*, Vol. 15, No. 1, pp.39–68.

Sampigethaya, K., Li, M., Huang, L. and Poovendran, R. (2007) 'AMOEBA: robust location privacy scheme for VANET', *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 8, pp.1569–1589.

Schoch, E., Kargl, F., Weber, M. and Leinmuller, T. (2008) 'Communication patterns in VANETs', *IEEE Communications Magazine*, Vol. 46, No. 11, pp.119–125.

Simplot-Ryl, D., Stojmenovic, I. and Wu, J. (2005) 'Energy-efficient backbone construction, broadcasting, and area coverage in sensor networks', in I. Stojmenovic (Ed.): *Handbook of Sensor Networks*, pp.343–380.

Steinbrecher, S. and Kopsell, S. (2003) 'Modelling unlinkability', in *Privacy Enhancing Technologies*, pp.32–47.

Stojmenovic, I., Ruhil, A.P. and Lobiyal, D.K. (2006) 'Voronoi diagram and convex hull based geocasting and routing in wireless networks', *Wireless Communications and Mobile Computing*, Vol. 6, No. 2, pp.247–258.