

Evaluation of the Use of Guard Nodes for Securing the Routing in VANETs

Juan A. Martinez, Daniel Viguera, Francisco J. Ros, and Pedro M. Ruiz

Abstract: We address the problem of effective vehicular routing in hostile scenarios where malicious nodes intend to jeopardize the delivery of messages. Compromised vehicles can severely affect the performance of the network by a number of attacks, such as selectively dropping messages, manipulating them on the fly, and the likes. One of the best performing solutions that has been used in static wireless sensor networks to deal with these attacks is based on the concept of watchdog nodes (also known as guard nodes) that collaborate to continue the forwarding of data packets in case a malicious behavior in a neighbor node is detected. In this work, we consider the beacon-less routing algorithm for vehicular environments routing protocol, which has been previously shown to perform very well in vehicular networks, and analyze whether a similar solution would be feasible for vehicular environments.

Our simulation results in an urban scenario show that watchdog nodes are able to avoid up to a 50% of packet drops across different network densities and for different number of attackers, without introducing a significant increase in terms of control overhead. However, the overall performance of the routing protocol is still far from optimal. Thus, in the case of vehicular networks, watchdog nodes alone are not able to completely alleviate these security threats.

Index Terms: Routing, security, vehicular ad hoc networks (VANETs).

I. INTRODUCTION

Vehicular ad hoc networks (VANETs) have emerged with a great strength gaining a lot of interest by government, traffic authorities, car manufacturers, and operators due to the different possibilities that they offer related to the car industry. They open a new market that allows for different services like enhanced safety, traffic congestion detection and avoidance, and Internet connectivity, among others.

Many of the envisioned VANET services require an effective solution to send data messages to vehicles located farther than their radio range. For this reason, multi-hop routing protocols are needed to deliver such messages to their intended destination. Most VANET routing protocols in the literature [1], [2] like geographic source routing (GSR) [3], spatially aware routing (SAR) [4], anchor-based street and traffic aware routing (A-STAR) [5], geographical opportunistic routing for vehicular networks (GeOpps) [6], or beacon-less routing algorithm for vehicular environments (BRAVE) [7] are based on geographical routing to reach the destination. Nevertheless, some of them also

employ additional information like traffic density (A-STAR) or the trajectory of vehicles (GeOpps). BRAVE, on its hand, ensures that the next selected forwarder is able to receive the data packet successfully. It has shown a great performance in terms of delivery ratio with respect to the other approaches. BRAVE is a beaconless routing protocol that uses an opportunistic forwarding mechanism. After receiving a packet from a sender, neighboring nodes propose themselves as candidate forwarders depending on their position.

However, most of the proposed routing solutions for VANET (including BRAVE) have not considered security issues. This is, they are not able to deal with certain situations such as spoofing, sybil attacks, selective forwarding, or sinkhole attacks, where malicious nodes try to impair the routing protocol by not forwarding the information to other nodes. For this reason, a variety of solutions to these attacks were proposed in the literature [8]–[10].

The IEEE 1609.2 standard [11] for securing wireless access in vehicular environments (WAVE) addresses the issues of securing vehicle-to-vehicle (V2V) communication against spoofing and eavesdropping by using a public key infrastructure (PKI). Thus, a certification authority (CA) will be responsible of generating and managing digital certificates. This standard proposes that vehicles sign the messages and piggyback the certificate of the sender, which contains the corresponding public key. When the destination receives the message, it will be able to validate the authenticity and integrity of the message. If every V2V message includes this public key, normally as a X.509 certificate, their size will be notably increased.

In this paper, we extend BRAVE by introducing guard nodes giving birth to our proposal secure-BRAVE called S-BRAVE. In S-BRAVE, messages are signed by taking advantage of the PKI. Nevertheless, we have developed an efficient certificate exchange mechanism where the certificate will be inserted in V2V messages only if the other vehicle has not received it yet. Thus, authenticity and integrity are guaranteed for every message transmitted along the VANET. Since our target in this paper is in how to secure the routing protocol we have not dealt with the issue of certificates revocation. However, we have worked in this issue in a previous paper already published where we exploited the capabilities of the next generation networks (NGN) to do it [12].

On the other hand, in the environment of wireless sensor networks (WSNs), a proposal for strengthening the security capabilities of a routing protocol have obtained very good results in terms of packet delivery ratio (PDR). This proposal is based on the concept of watchdog nodes or guard nodes, neighboring vehicles that watch packet exchanges to ensure that the packet is forwarded by the intended next hop [13]. We have applied this technique in the VANET environment making BRAVE able

Manuscript received July 27, 2012.

The authors are with the Department of Information and Communications Engineering, University of Murcia, Spain, email: juanantonio@um.es, sothek@gmail.com, {fjros, pedrom}@um.es.

Digital Object Identifier 10.1109/JCN.2013.000025

to transmit messages in hostile scenarios where malicious nodes selectively forward messages in order to cause packet losses. For this purpose, using the aforementioned technique, neighbors watch other selected nodes to be sure that they forward packets to the next hop. If a node is selected to forward the packet and it does not transmit it, then neighboring nodes will select themselves as forwarders, taking the responsibility of sending the packet to the next hop. The whole process is detailed in later sections.

The remainder of this paper is organized as follows. In Section II, we present the related work. Section III describes the BRAVE routing protocol as well as the main threats that it must deal with. In Section IV, we detail S-BRAVE, our proposal. The evaluation of the performance of S-BRAVE is shown in Section V. Finally, we summarize the main outcomes of this protocol and conclude the paper in Section VI.

II. RELATED WORK

Raya and Hubaux [14] detail security mechanisms that should be applied to vehicular networks in order to protect them. They review different aspects of security in this environment, such as authenticity, privacy, different kinds of attacks, use of certificates, and so on. However, they do not discuss these problems from the point of view of a routing protocol. That is, they describe only appropriate security mechanisms, but not how to use them in an efficient way taking into account the overhead that these mechanisms would introduce into the routing protocol.

Papadimitratos *et al.* [15] deal with the problem of securing beacon messages. They propose to sign them attaching also the sender certificate into them. Geocast dissemination messages are also digitally signed and augmented with the certificate of the sender. Nevertheless, attaching always the certificate of the sender increases packet size. Given that both beacons and data messages are enlarged, collision probability increases, thus reducing reliability and increasing the number of retransmissions.

On the other hand, the problem of securing a VANET routing protocol is analyzed in [16] and [17]. The security extension is applied to position-based routing protocol (PBR), an approach developed within the context of the network on wheels (NoW) project. PBR, after knowing the position of the destination, uses a greedy forwarding algorithm to reach it. Nevertheless, in urban scenarios streets constrain vehicles' movements, so the greedy process will often reach local optima where no neighbor will provide advance to the destination. BRAVE solves this by introducing the next junction as a first destination to be reached into the first packet to be transmitted, which is the packet containing the data information to be sent.

Among the security techniques included in PBR, there is an aspect which is worth highlighting. For packets to be delivered through more than one hop, they introduce two signatures, one for the source node, and another one for the sender (hop by hop). Thus, each packet will be signed twice, and after verifying these signatures, in each hop, the sender's signature will be removed introducing a new one corresponding to the next hop. By contrast, our proposal only introduces one signature in each packet. The first packet sent by BRAVE will be forwarded as transmitted by the source, without resigning it. On the other hand, the rest

of protocol control messages, which are only one-hop messages, will be signed by the node that sends them.

III. BACKGROUND

A. BRAVE

BRAVE is a geographic routing algorithm which does not consider information gathered from periodic beacons in making routing decisions. In this way, several problems related to dealing with stale neighbor information [18] are avoided. This approach provides a better adaptation to network topology changes and lower overhead than other algorithms. The idea comes up from the beacon-less on demand strategy for geographic routing in WSNs (BOSS) algorithm [19], originally developed for WSNs.

In BRAVE, there are four message types: DATA, RESPONSE, SELECT, and ACK. Due to the existence of many data sources in the network, every message includes a unique key which is the result of concatenating the identifier of the source node and a sequence number. BRAVE uses an opportunistic scheme when making forwarding decisions. Thus, when a node intends to send data to a destination, it broadcasts the DATA packet (scheduling also a timer in case no neighbors answer the message). After receiving this DATA packet, every neighboring node schedules a response timer before answering. The more progress provided by a neighbor towards the destination, the less such neighbor has to wait to answer with a RESPONSE message.

The sender selects the neighbor whose RESPONSE message arrives first. For that, a SELECT message aimed at the chosen vehicle is broadcasted. Therefore, all neighbors get aware of the vehicle that has been selected to be the next forwarder. Those which are not the final destination nor the next forwarder, delete the DATA message from their buffer and go back to the initial state. The selected vehicle, on its hand, after receiving the SELECT message becomes the current forwarder and starts over the forwarding process by broadcasting the DATA message. Such message is expected to reach the previous hop, acting as an implicit acknowledgment of reception. Otherwise, the previous hop rebroadcasts the DATA message and the already selected forwarder answers with an explicit ACK message. On the other hand, if the forwarding node observes that it has no neighbors to which forward the message, then it stores the message in its buffer. An explicit ACK message is sent in this case too. Thus, the vehicle carries the message until it receives a new beacon indicating that there is a neighbor. The reception of this beacon triggers a new event that makes the node check whether there are messages to be delivered. In such case, the whole process starts over.

The former mechanism has been shown to have better performance [7] than other beacon-based algorithms because, instead of being the sender the one that selects the next forwarder based on information acquired via beacons (which could not be up to date), the neighbors propose themselves as good forwarders after receiving the DATA packet. In addition, BRAVE also takes advantage of maps by introducing the next junction of the street as a first destination of the message, besides the final destination. Once this next junction is reached, BRAVE determines a

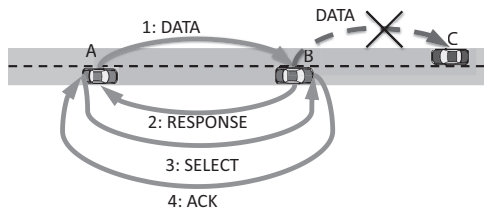


Fig. 1. Selective forwarding / sinkhole attack performed by vehicle *B*.

new one closer to the final destination. This process is repeated until the message is received by the intended destination.

B. Routing Threats

There are different security threats depending on the layer that they are aimed at. Focusing on the network layer, attacks like black hole, selective forwarding, wormhole, and the likes are described in the literature. Depending on the messages exchanged in routing protocols, some of them are more vulnerable to these attacks than others. Thus, it is important to analyze the routing protocol to find the threats that affect it the most.

In BRAVE, the first packet to be transmitted contains the data information, and only nodes that receive this message participate in the next hop selection mechanism. Hence, a black hole attack consisting of a malicious node that silently discards or drops messages without informing the source that the data did not reach its intended recipient will not affect BRAVE at the time of selecting a new neighbor. However, an attacker might participate on the exchange of BRAVE messages and, once it holds the DATA packet and sends back an ACK, it could stop forwarding prematurely. Fig. 1 illustrates this case.

BRAVE messages are not authenticated nor integrity-protected, enabling other kinds of attacks by a malicious node. Thus, it can manipulate the information stored in the message, for instance changing the destination of the packet or altering its content. This issue can be alleviated by employing a PKI, so that vehicles will be able to sign data packets with their private keys. Hence, receivers can validate packets by using the public key contained within the digital certificate of the sending vehicle. In the following section, we describe the mechanism employed to exchange these certificates among nodes.

In addition, malicious nodes can attack BRAVE by proposing themselves as the best candidates to forward the packet to the destination. To do so, they will take advantage of the timer scheduling, by which the node that answers first to the DATA packet is selected as the next hop. Thus, by answering first, a malicious node can manage to get elected as next hop.

Another chance for malicious nodes to harm BRAVE is by not issuing the SELECT message once that they have sent the DATA packet. In this way, no neighbor will be selected as a relay.

Other more elaborated attacks, like the sybil attack or wormhole attack, can also be practiced within the VANET environment. In a sybil attack, a malicious node presents multiple identities with different locations to other vehicles in the network. This attack is more sophisticated than the previous ones because, in this case, the malicious node announces itself also in other locations, taking advantage of these positions to be selected as the

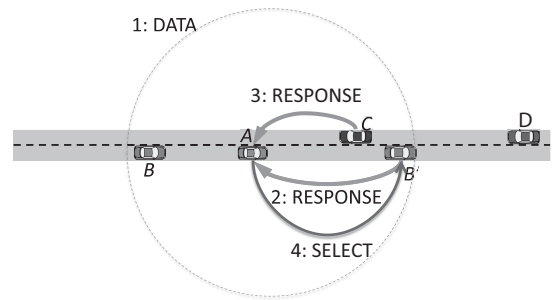


Fig. 2. Sybil attack.

best neighbor to forward a packet. For instance, in Fig. 2 vehicle *B* creates a new identity *B'* in a more advantageous location. Hence, it is selected as the best forwarder to the destination.

The only way for a malicious node to create more than one entities is to have more than one pair of public/private keys. There are different alternatives for it, like the use of pseudonyms or installing several certificates within the vehicle. We simplify the problem by forcing a single certificate per vehicle, which is generated by a trusted CA. In such case, the sybil attack gets reduced to its minimum exponent. That is, a vehicle could forge its position, but could not create multiple identities.

Finally, a wormhole attack requires the cooperation of at least two malicious nodes. It consists of two vehicles that create a tunnel between them, so that they can forge their distance to the destination. For instance, if the malicious nodes are far from each other more than one hop, by using the tunnel, for the rest of the neighbors it would be as if there were no distance between them. This attack is harder to perform because of the high variability of links among neighboring nodes due to the high speed of the vehicles.

IV. SECURING THE BRAVE PROTOCOL

In this section, we develop S-BRAVE, an extension of the BRAVE routing protocol targeted at addressing the security threats that have been previously detailed.

In first place, we will provide authentication and integrity by exploiting a PKI. Thus, the source vehicle signs data packets with its private key and the receiver uses the public key of the sender to check the validity of the packet. Since the receiver node requires the sender certificate, it is necessary a previous exchange (introducing extra overhead). We propose a certificate exchange mechanism that tries to reduce the associated overhead. It is described in subsection IV-A.

Although the use of a PKI is a building block of our protocol, it is not enough to avoid the threats discussed in subsection III-B. Therefore, in subsection IV-B, we detail the modifications we have done to the behavior of the original BRAVE protocol to deal with malicious nodes. Such additions include the use of guard nodes.

A. Certificate Exchange

Since a VANET is a distributed environment, vehicles must trust each other somehow. They cannot always access the infrastructure to check the validity of the messages. Therefore,

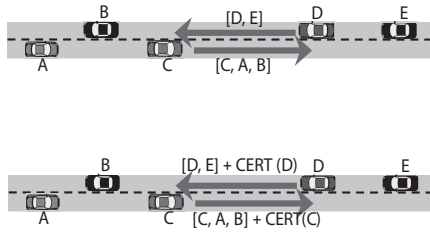


Fig. 3. Certificate exchange via periodic beacons.

the most appropriate way to check the authenticity is to sign them with a private key, allowing other nodes to gather the corresponding public key to verify that signature. In S-BRAVE, we assume a unique CA which is the same for all the vehicles in the VANET.

Thanks to the PKI, each vehicle owns a unique identifier and a pair of keys (public and private) as well as a certificate issued by the CA. Besides, with a single CA, vehicles can easily check the validity of the certificate associated to a message sent by any vehicle of the network. That is, a vehicle does not need to ask any third-party.

The first problem to deal with is how to exchange certificates among vehicles. Every time a node receives a message, it must have the certificate of the sender node in order to authenticate it and to check the integrity of the message. One option is to include the certificate in every data message. However, this is not efficient because certificates have a big size and they would increase the overhead.

We propose a reactive certificate exchange method which minimizes the number of certificate exchanges. Every beacon sent will include a cache of known neighbors, being a known neighbor one whose certificate is stored within the vehicle. When a vehicle receives this beacon, just by looking for its own identifier in the neighbor list, it will be able to determine if its certificate is present in the cache of the neighbor. Such cache is updated with a less recently used (LRU) scheme. If the certificate identifier is not present, then the vehicle will include its own certificate in the next beacon round. Using this strategy only the first beacon will include the certificate, the following messages between those vehicles will not need to include certificates for validation. Besides, other nodes that receives a beacon with the certificate can take advantage of this exchange method to store the certificate for possible use in the future. Fig. 3 shows this exchange of messages.

Given that certificates are exchanged in advance, it is possible to authenticate routing messages (RESPONSE, SELECT, and ACK) by just using digital signatures. However, in order to check the validity of a DATA message, a vehicle located farther than one hop of the sender needs a mechanism to get the certificate of the source. The reason is that DATA messages are signed by the source, but not by intermediate relays.

Our proposal to solve this problem is based on modifying RESPONSE and SELECT messages. A bit included in the RESPONSE message will indicate if the responding vehicle needs the certificate of the source. After receiving this RESPONSE, the SELECT message will be extended with the certificate of the source node depending on this bit (see Fig. 4). This protocol

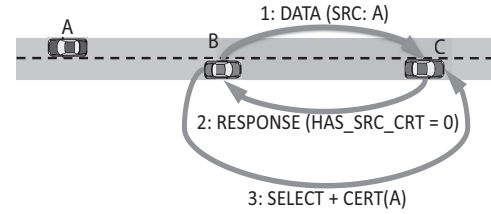


Fig. 4. Certificate exchange of the source vehicle.

modification entails an overhead decrease mainly when the path between source and destination is stable.

The CA is able to add malicious nodes to a certificate revocation list (CRL), so that their messages become invalid for the other vehicles. Thus, when a node detects a malicious node it can notify the CA to update its CRL by using an infrastructure network. This way, after receiving the updated CRL, nodes will discard the packets whose source node matches one of those identities listed in the CRL. As commented in the introduction of this paper, the issues related to distribute and update CRLs are discussed in [12].

B. S-BRAVE Operation

The certificate exchange scheme described before is a basic building block of our solution. However, BRAVE is still weak against a selective forwarding attack that can be accomplished in two ways. In the first one, a malicious node does not continue the forwarding of the DATA message, nevertheless, it answers its previous hop with an ACK message making it believe that it has forwarded it. In the second one, the malicious node does not send the SELECT message. For instance, if a node starts the exchange of messages but it does not send the SELECT message, there will not be a forwarding node and, therefore, the message will not be forwarded. In both cases, the previous hop may think that the forwarding was completed.

In order to try to avoid this type of situation, S-BRAVE employs the concept of watchdog nodes or guard nodes in the following way. Every neighboring vehicle that provides advance to the destination will act as a guard node. Those vehicles not selected as the next forwarder will try to ensure that the whole DATA forwarding process is completed. They keep on listening to the next forwarder, checking whether it retransmits the DATA message. If a guard node does not receive this message, it will take the role of the next forwarder by taking the responsibility of sending the DATA message to the next hop. They also include the detected malicious vehicle in a black list, to avoid that it gets selected as the next forwarder in the future.

We have modified the original BRAVE protocol as indicated next.

First of all, we have modified the ACK message. A new bit has been added, which indicates the reason why this ACK has been sent. Thus, a vehicle can send this message by two reasons: DATA message has already been forwarded previously, or it has been buffered by the vehicle because it did not have any neighbors which provided advance towards the destination.

In addition, we have also defined a black list where neighbors which do not forward messages are registered into. This mechanism is used to avoid a malicious node to continuously

impair the protocol performance by being selected by the same node one time after another. Thus, guard nodes will ignore the messages coming from a node of the black list. For instance, after a node sends an answer with a RESPONSE message, the neighbors that have this node into their black list, will also send their RESPONSE message instead of canceling their timers. Besides, the sender of the DATA message will also ignore the RESPONSE of a node if its identifier is stored in the black list.

Finally, neighboring vehicles that receive a RESPONSE or SELECT message do not go back to the initial state. Instead, they will keep the DATA message just received, watching for the right exchange of messages and the subsequent DATA message forwarding by the selected node. They also schedule a timer that waits for this exchange to be succeeded within a period of time, otherwise the guard nodes will come to the conclusion that a malicious node is attacking by preventing the packet from being delivered. In such case, they collaborate to forward the DATA packet.

In the following, we detail S-BRAVE and provide some pieces of pseudocode of the most relevant operations that must be performed.

The sender vehicle, after issuing a DATA packet, schedules a timer waiting for responses from neighboring nodes (*awaitingRESPONSE*). This packet is the one that triggers the next hop selection. Procedures 1–4 deal with main message exchanges of S-BRAVE. In addition, Procedure 5 defines what vehicles do after their timers expire.

Procedure 1 processDATA (m:Message, src:Address, dst:Address)

```

1: if (noActiveTimers) then {Node receives DATA in initial state}
2:   if (dst == ownAddress) then {Node is the destination of DATA}
3:     send(RESPONSE);
4:     scheduleTimer(awaitingToSELECT);
5:   else if (nodeProvidesAdvanceToDest(dst)) then
6:     scheduleTimer(awaitingToAnswer);
7:   end if
8: else if ((src == selectedNode) && awaitingACK) then
9:   exit; {Next hop, i.e., selectedNode, retransmit the packet}
10: else if (awaitingForwardedMsg) then {guard nodes}
11:   cancelTimer(awaitingForwardedMsg);
12:   if (nodeProvidesAdvanceToDest(dst)) then
13:     scheduleTimer(awaitingToAnswer);
14:   end if
15: end if

```

In Procedure 1, a vehicle that has received a DATA message can be in two states. The first one is the *idle* state, where the vehicle is at the beginning of processing the DATA message. If the node is the final destination of the packet it will immediately answer with a RESPONSE message, also scheduling a timer to receive the SELECT message. Otherwise, only the vehicle providing advance to the destination will schedule a timer to answer with a RESPONSE message. The receiver can also be in the *awaitingACK* state, meaning that it has nearly finished the exchange of messages but it is expecting the ACK message. After

receiving the ACK, the node would go back to the *idle* state. Finally, if the node is a guard node and receives this DATA message it will cancel its timer of watching the packet, scheduling a new timer that depends on the progress provided with respect to the destination.

Procedure 2 processRESPONSE (m:Message, src:Address, dst:Address)

```

1: if (awaitingRESPONSE && (dst == ownAddress)) then
2:   cancelTimer(awaitingRESPONSE);
3:   send(SELECT, src);
4:   selectedNode ← src;
5:   scheduleTimer(awaitingACK);
6: else if (awaitingToAnswer) then
7:   cancelTimer(awaitingToAnswer);
8:   scheduleTimer(awaitingNextForwarderSelected);
9: end if

```

When a vehicle receives a RESPONSE message (Procedure 2), it will send back to the most promising forwarder a SELECT message, also scheduling a new timer. On the other hand, if the vehicle is not the best forwarder, it will schedule a new timer to watch the messages exchange to act as a guard node.

Procedure 3 processSELECT (m:Message, src:Address, dst:Address)

```

1: if (awaitingSELECT) then
2:   cancelTimer(awaitingSELECT);
3:   if (finalDest == ownAddress) then
4:     send(ACK,src);
5:     scheduleTimer(awaitingPostProc);
6:   else if (dst == ownAddress) then
7:     if (noNeighbors) then
8:       send(ACK); {It buffers the DATA}
9:     else
10:      send(DATA);
11:      scheduleTimer(awaitingRESPONSE);
12:    end if
13:   else
14:     scheduleTimer(awaitingForwardedMsg);
15:   end if
16: else if (awaitingToAnswer) then
17:   cancelTimer(awaitingToAnswer);
18:   scheduleTimer(awaitingForwardedMsg);
19: else if (awaitingNextForwarderSelected) then
20:   cancelTimer(awaitingNextForwarderSelected);
21:   scheduleTimer(awaitingForwardedMsg);
22: end if

```

Procedure 3 describes what happens when a vehicle receives a SELECT message. If it has already sent a RESPONSE message, it will be selected as the next forwarder. Thus, it will cancel its waiting timer (*awaitingSELECT*). In case the vehicle is the final destination, it will send an ACK message back to the previous hop. Otherwise, it will broadcast the DATA message unless it will not have any neighbors around it. In this latter case, it will store the message in a buffer, answering with an ACK

which specifies this. Guard nodes will cancel their timers and will schedule new ones because the messages exchange is being performed correctly.

Procedure 4 processACK (m:Message, src:Address, dst:Address)

```

1:  if (awaitingACK) then
2:    cancelTimer(awaitingACK);
3:    exit; {Node goes back to initial state}
4:  else if awaitingForwardedMsg then
5:    if (m.reason == Forwarded) then {reason is an attribute
      of the message m}
6:      cancelTimer(awaitingForwardedMsg);
7:      send(DATA);
8:      scheduleTimer(awaitingRESPONSE);
9:    else {m.reason == Buffered}
10:     if (noPromisingNeighbors) then
11:       buffer(DATA);
12:     else
13:       cancelTimer(awaitingForwardedMsg);
14:       send(DATA);
15:       scheduleTimer(awaitingRESPONSE);
16:     end if
17:   end if
18: end if

```

Procedure 4 describes the ACK reception process. If the vehicle that receives the ACK is the sender, it will cancel its timer assuming the whole messages exchange is completed. On the other hand, guard nodes will analyze the reason of sending this ACK. In case the message indicates a forwarding not heard by them, they will take the role of forwarders by broadcasting the DATA packet.

Procedure 5 timerExpires(timer)

```

1:  if (timer == awaitingToAnswer) then
2:    send(RESPONSE);
3:    scheduleTimer(awaitingSELECT);
4:  else if (timer == awaitingPostProc) then
5:    exit; {Node goes back to initial state}
6:  else if (timer == awaitingNextForwarderSelected) then
7:    send(DATA);
8:    scheduleTimer(awaitingRESPONSE);
9:  else if (timer == awaitingForwardedMsg) then
10:   send(DATA);
11:   scheduleTimer(awaitingRESPONSE);
12: end if

```

In Procedure 5, if the vehicle state is *awaitingToAnswer*, it will send a RESPONSE message. This is the case where the vehicle has received the DATA packet and has scheduled a timer to answer to it. On the other hand, guard nodes (the last two cases) will take the role of new forwarders by broadcasting the DATA message.

In the remainder of this section, we analyze possible attacks and how S-BRAVE behaves against them.

Let us start with a way of selective forwarding attack in which a malicious node does not forward the DATA message. The

neighbors which provide advance to the destination, after receiving the DATA packet, will trigger a timer before sending their response (line 6, Procedure 1). The one which provides the highest progress towards the destination will answer first with a RESPONSE message (line 1, Procedure 5). However, the rest of these neighbors schedule a new timer waiting for a vehicle to be selected as the next forwarder (line 1, Procedure 2). After receiving this SELECT message, guard nodes cancel their timer, scheduling a new one to be sure that this new forwarder will deliver the message to the next hop (lines 19–22, Procedure 3). If any of the aforementioned timers expire, guard nodes will assume that the vehicle selected to forward the message is a malicious one. Hence, they will select themselves as new forwarders, taking the responsibility of sending the message to the next hop (lines 6–12, Procedure 5). Not all timers expire at the same time. The vehicle whose timer expires first will start sending the DATA message. In order to reduce the overhead of the protocol, the other guard nodes will cancel the sending of this DATA when they overhear the DATA from another guard node. Depending on their relative positions to this new forwarder will act as guard nodes, scheduling a new timer, or just going back to the initial state.

On the other hand, if the malicious node replies with an ACK message, it will have to select a reason for it as any node. If the guard nodes receive an ACK message with the reason of *message already forwarded* they will react by sending the DATA message (line 5, Procedure 4). However, if they receive an ACK with the *message buffered* reason, they will check their neighbor list to have an idea of how many neighbors there are around. If there are any other neighbors providing advance to the destination apart from the vehicle which sent the ACK, they will take the responsibility of sending the DATA message. In this process, the aforementioned mechanism to reduce protocol overhead by overhearing DATA transmissions take place. Otherwise, guard nodes will buffer the packet until new neighbors come close to them (line 9, Procedure 4).

Thus, S-BRAVE is able to deal with the selective forwarding attack as well as providing integrity and authenticity to the messages. Any attacker can pretend to be the best forwarding node but the attack will not be successful if there are surrounding guard nodes. Packet identifiers are unique, so although more than one guard node would detect the attacker and therefore would forward the packet, duplicate packets will converge in the next hops. So, they can be detected and avoided.

Another way of the selective forwarding attack occurs when a malicious node that has been selected as next forwarder goes on with the entire process to deliver the DATA to the next hop, but it does not allow any other neighbor to be selected by not sending a SELECT message. Thus, no other neighbor will receive the confirmation to forward the packet to the destination. S-BRAVE, in order to counterattack this situation, uses the *awaitingNextForwarderSelected* and *awaitingForwardedMsg* timers. Thus, when these timers expire, a guard node will take the role of new forwarders starting the protocol to deliver the message to the next hop (lines 6 and 10, Procedure 5).



Fig. 5. Map of Murcia city center and access roads used in our simulations.

V. PERFORMANCE EVALUATION

We have compared both protocols BRAVE and S-BRAVE within the network simulator ns-2, version 2.33¹. We consider a 5×4 km² scenario which consists of the main access roads and streets of the city center of Murcia, Spain (see Fig. 5). It contains 53 streets and 28 junctions. This map, as well as the vehicular mobility patterns, have been generated by means of the well-known simulation of urban mobility (SUMO) road traffic simulator².

Vehicles move through 20 predefined routes at a maximum speed of 50 km/h inside the city, and 80 km/h on the highway that crosses the scenario during 885 seconds. The routes followed by the vehicles have been selected according to realistic situations. We have also considered a wide range of traffic densities. Vehicles are injected into their routes at a certain traffic rate. This rate is varied from 1/45 to 1/15 vehicles per route per second. Thus, a $1/x$ rate means that each x seconds a new vehicle is injected into its route.

In our simulations, wireless signals propagate according to the two-ray-ground model. Vehicles carry out their communications via an 802.11p interface card, implementing the enhanced ns-2 802.11 physical and medium access control (MAC) models [20]. The transmission power is adjusted to allow a maximum transmission range of 250 m. Within this scenario we have simulated 10 runs for each configuration, each of them with different traffic sources randomly selected for this purpose. Therefore, figures in this section show the average of such runs along with their corresponding 95% confidence intervals. Since our interest in this paper is to analyze the effects of the attacks and whether the guard node scheme can deal with such security threats, we

have considered a simple propagation model to avoid introducing more bias in the experiment. For our experiments, we do not require detailed radio propagation models such as the one in [21].

A. BRAVE vs. S-BRAVE

We have compared both protocols for a varying percentage (0%, 5%, 10%, and 15% of the total number of vehicles) of malicious nodes that randomly apply one of the two ways of the selective forwarding attack already explained. Regarding the cache of known neighbors, for our simulation, we have used an unlimited cache where all the neighbors met along the road by a vehicle will be added to the cache. This way, if a vehicle meets an old neighbor, it will have stored its certificate making our protocol to obtain its best possible results despite its introduced overhead.

Fig. 6(a) shows the performance of both approaches in terms of the PDR. The x-axis represents the different simulated densities, while the y-axis shows the PDR obtained in a scenario where there are not any malicious node. In this scenario, both approaches obtain great results with more than 80% of the packets being delivered to the destination. Analyzing the figure in more detail, we can see that the performance of S-BRAVE is lower than BRAVE for sparse scenarios. This is caused by the false positives occurred during the simulation and their corresponding overhead. During the simulation, guard nodes watching the packets to be forwarded do not receive the forwarded message, making the decision of being themselves the new forwarders. However, the denser the scenario the better performance is obtained from S-BRAVE, reaching the same results as BRAVE (and even outperforming it).

As the percentage of malicious nodes is increased, the performance of both protocols is deteriorated. Taking a look at Fig. 6(b), where 5% of the vehicles are malicious, S-BRAVE managed to deliver from 40% to 70% of the packets to their destination. However, BRAVE is only able to deliver from 10% to 30%. S-BRAVE outperforms BRAVE in at least 20%. This gap is even higher as the density of the scenario is increased, reaching a gap of 50% of the PDR.

This improvement is also shown in Figs. 6(c) and 6(d), where the percentage of simulated malicious nodes is 10% and 15%, respectively. In these figures, S-BRAVE outperforms BRAVE in terms of PDR with a gap of at least 20% in the former and 10% in the latter. However, it is worth mentioning that BRAVE is only able to deliver up to 13% and 10% of packets with 10% and 15% of malicious nodes in the best case. On its hand, S-BRAVE is able to deliver between 40% and 45% of messages with a high density scenario. The reason for this is that when density is low, even though S-BRAVE manages to deal with attackers, it may happen that the attacker is the only forwarding alternative. When density is higher, it is easier to find guard nodes that can help avoid attacks.

Regarding the communications delay, Figs. 7(a)–7(c) show the delay for the different simulated densities with a 5%, 10%, and 15% of attackers, respectively. As the percentage of malicious nodes increases, it is more difficult to find nodes which can act as guards, hindering the routing process due to the lack of nodes that take the responsibility of delivering the packets.

¹<http://www.isi.edu/nsnam/ns/>

²<http://sumo.sourceforge.net/>

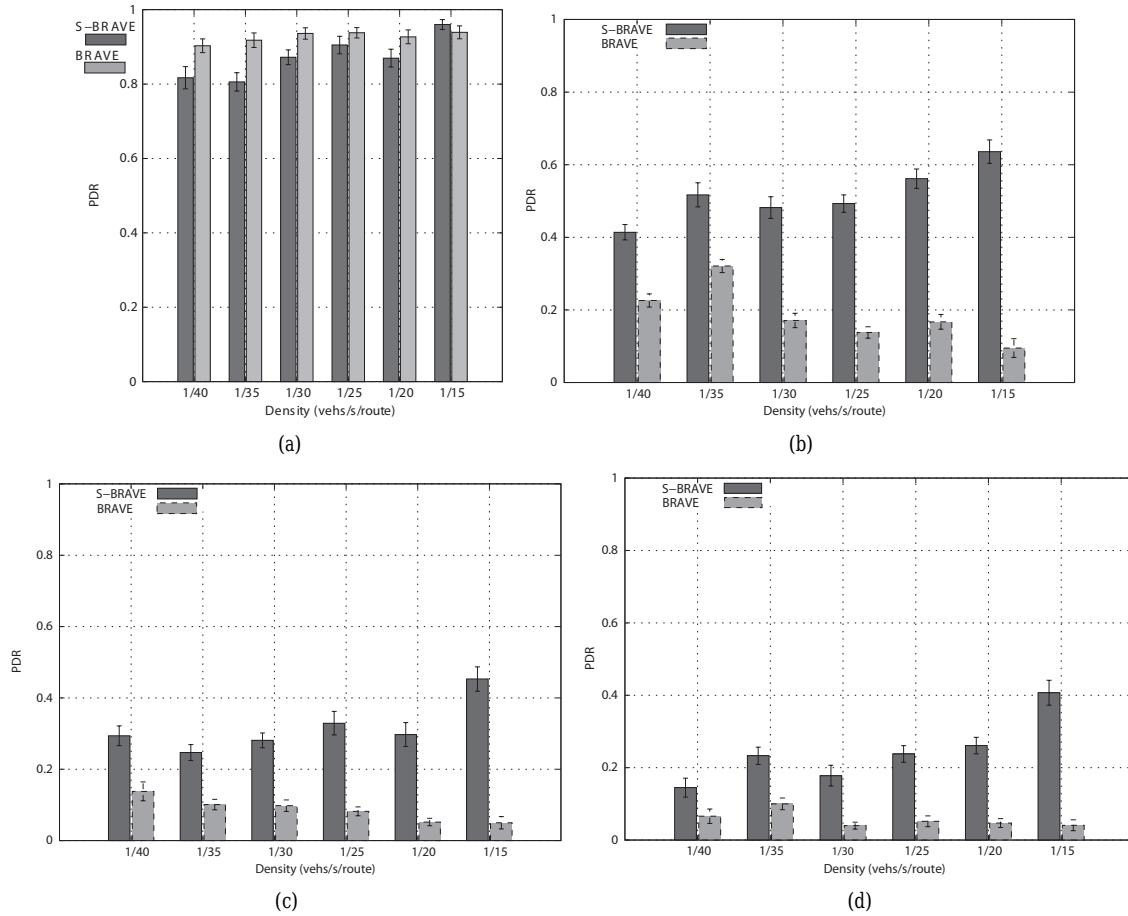


Fig. 6. Percentage of PDR for 0% , 5%, 10%, and 15% of malicious nodes: (a) PDR for 0% malicious nodes, (b) PDR for 5% malicious nodes, (c) PDR for 10% malicious nodes, and (d) PDR for 15% malicious nodes.

Therefore, vehicles more often reach a local optimum having no trustable neighbor that provides advance. In that situation, vehicles make the decision of storing the packet until they find a suitable neighbor. This is the reason why as the percentage of attackers is increased the delay increases.

Despite these facts, BRAVE usually obtains a lower delay compared with S-BRAVE in high density scenarios. Looking back at PDR graphs, BRAVE for those densities was hardly able to deliver up to 20% of the packets. Matching these results, we can deduce that BRAVE is only able to deliver packets if the senders are near the destinations. If they are distant, it is more likely that it reaches a malicious node that drops the message.

In order to provide more insight onto the performance of the protocols, we also compute the overhead per delivered message, the overhead per delivered message per hop, and the number of delivered packets against the number of hops they have gone through.

As expected, S-BRAVE has more overhead per successful delivery than BRAVE (see Fig. 8(a)). In fact, S-BRAVE sends certificates when needed while BRAVE does not use it. However, if we consider that overhead per successful delivery per hop, we can see Fig. 8(b) that S-BRAVE only adds little overhead compared to BRAVE, despite the need of certificates. The reason for the higher overhead in Fig. 8(a) is that S-BRAVE manages to deliver packets to destination which are located far from the

source (# of hops), while BRAVE just can not do it.

This is corroborated by Fig. 8(c), where we have related the number of delivered packets by both protocols with the distance (in hops) they are able to reach. For this purpose we have contemplated, not the average value of the 10 runs of the scenario but the total amount of delivered packets of these runs. In light of these results, we can see that S-BRAVE delivers more messages than BRAVE, and it manages to deliver them even to vehicles located many hops away from the source.

Despite the gap between BRAVE and S-BRAVE regarding the PDR, S-BRAVE still suffers up to a 40% of yield loss in low dense scenarios where a vehicle has few neighbors able to forward the messages towards the destination, compared with a scenario without malicious nodes. This means that with just these mechanisms, we are still far from a secure routing protocol able to countermeasure every attack of malicious nodes.

VI. CONCLUSION

We analyze the problem of secure routing in VANET. In particular, we focus on the BRAVE routing protocol, which is one of the best performing proposals so far. This is a challenging problem because of the high mobility of the vehicles as well as because BRAVE also uses the store-carry-and-forward paradigm to make nodes act as ferries for the packets if there are

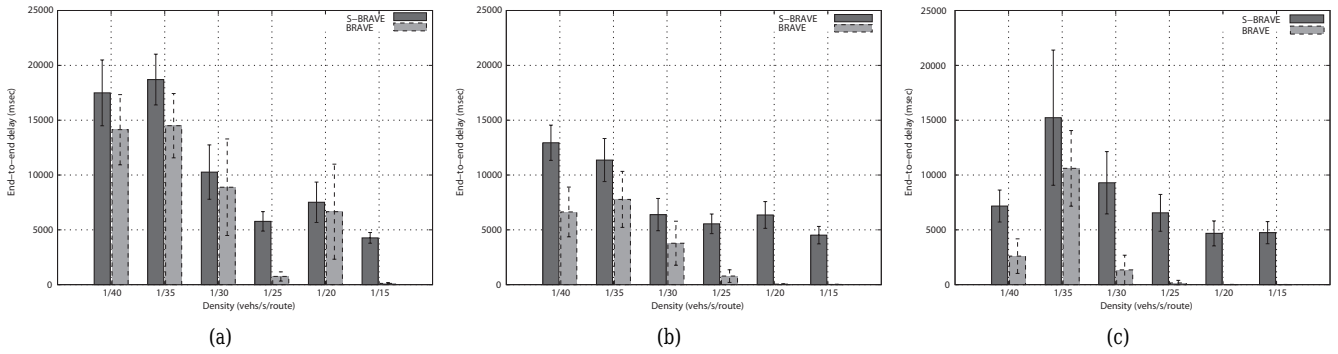


Fig. 7. Delay for 5%, 10%, and 15% of malicious nodes: (a) 5% of malicious nodes, (b) 10% of malicious nodes, and (c) 15% of malicious nodes.

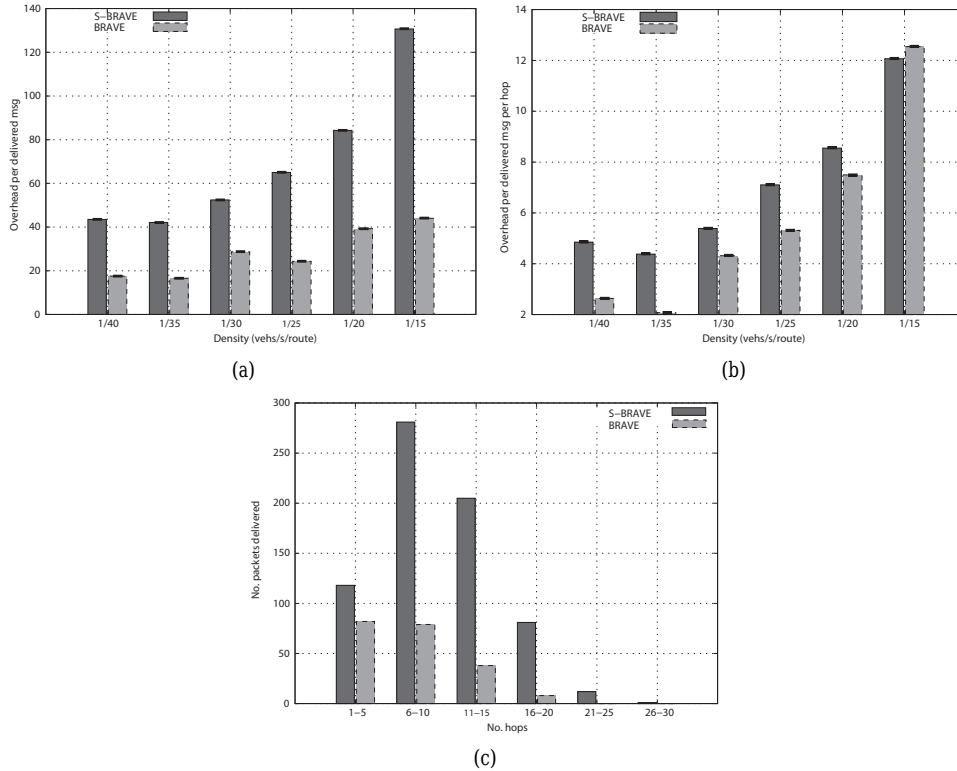


Fig. 8. Overhead (number of BRAVE messages) per hop (two first graphs) and number of delivered packets (last graph): (a) Overhead of both protocols for 5% malicious nodes per delivered msg., (b) overhead of both protocols for 5% malicious nodes per delivered msg. per hop, and (c) no. of delivered packets vs. the distance both protocols can reach.

no promising neighbors around the vehicle.

For this purpose, we have introduced a certificate exchange mechanism guaranteeing the authenticity and integrity of the messages as they traverse intermediate nodes until they reach their destination. Besides, we have also developed a way of securing BRAVE against selective forwarding attacks using neighboring nodes as guard nodes. They watch for the message to be sent by the next forwarder and, in case this vehicle does not forward the message, they take the responsibility of sending the message to the next hop.

In order to compare both protocols, we have implemented them in NS-2. In light of the results of the previous section, S-BRAVE outperforms BRAVE in terms of PDR. In spite of it, S-BRAVE is still far from a secure protocol where malicious nodes are not able to affect it. In low dense scenarios, S-BRAVE routing task is very arduous due to lack of neighbors. On the

other hand, in high dense scenarios, its performance gap compared with BRAVE is up to a 50% of the PDR. In addition, this enhancement of performance is achieved without significantly increasing the delay nor the overhead. However, S-BRAVE still suffers from the influence of the attackers obtaining maximum a PDR of about 50%.

REFERENCES

- [1] M. Mauve, A. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE netw.*, vol. 15, no. 6, pp. 30–39, 2001.
- [2] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Veh. Technol. Mag.*, vol. 2, no. 2, pp. 12–22, 2007.
- [3] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proc. IEEE IVS*, 2003, pp. 156–161.
- [4] J. Tian, L. Han, K. Rothermel, and C. Cseh, "Spatially aware packet rout-

- ing for mobile ad hoc inter-vehicle radio networks," in *Proc. IEEE ITSC*, 2003, pp. 1546–1551.
- [5] B. Seet, G. Liu, B. Lee, C. Foh, K. Wong, and K. Lee, "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications," in *Proc. IFIP-TC6 Netw.*, 2004, pp. 989–999.
 - [6] I. Leontiadis and C. Mascolo, "GeOpps: Geographical opportunistic routing for vehicular networks," in *Proc. WoWMoM*, 2007, pp. 1–6.
 - [7] P. Ruiz, V. Cabrera, J. Martinez, and F. Ros, "BRAVE: Beacon-less routing algorithm for vehicular environments," in *Proc. IEEE MASS*, 2010, pp. 709–714.
 - [8] J. Douceur, "The sybil attack," in *Proc. IPTPS*, 2002, pp. 251–260.
 - [9] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis & defenses," in *Proc. IPSN*, 2004, pp. 259–268.
 - [10] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc netw.*, vol. 1, no. 2–3, pp. 293–315, 2003.
 - [11] *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages*, IEEE. Std. 1609.2-2006, 2006.
 - [12] I. Lequerica, J. A. Martinez, and P. M. Ruiz, "Efficient certificate revocation in vehicular networks using NGN capabilities," in *Proc. IEEE VTC*, 2010, pp. 1–5.
 - [13] R. Marín-Pérez and P. M. Ruiz, "SBGR: A simple self-protected beaconless geographic routing for wireless sensor networks," in *Proc. IEEE MASS*, 2011, pp. 610–619.
 - [14] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *J. Comput. Secur.*, vol. 15, pp. 39–68, Jan. 2007.
 - [15] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: Design and architecture," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 100–109, 2008.
 - [16] C. Harsch, A. Festag, and P. Papadimitratos, "Secure position-based routing for VANETs," in *Proc. IEEE VTC*, 2007, pp. 26–30.
 - [17] A. Festag, P. Papadimitratos, and T. Tielert, "Design and performance of secure geo-cast for vehicular communication," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2456–2471, June 2010.
 - [18] V. Cabrera, F. Ros, and P. Ruiz, "Simulation-based study of common issues in VANET routing protocols," in *Proc. IEEE VTC*, 2009.
 - [19] J. Sanchez, R. Marín-Pérez, and P. Ruiz, "Beacon-less geographic routing in real wireless sensor networks," *J. Comput. Sci. Technol.*, vol. 23, no. 3, pp. 438–450, 2008.
 - [20] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in NS-2," [Online]. Available: http://dsn.tm.kit.edu/english/Overhaul_NS-2.php
 - [21] F. Martinez, M. Fogue, C. K. Toh, J. C. Cano, C. T. Calafate, and P. Manzoni, "Computer simulations of VANETs using realistic city topologies," *Wireless Pers. Commun.*, pp. 1–25, 2012.



Juan A. Martinez received his B.Sc. degree in Computer Science (2005), a Post-Graduate Diploma in New Technologies (2007) in Information and Communications Engineering, and M.S. degree in Advanced Information and Telematic Technologies (2009) in the Computer Science Faculty of the University of Murcia. He is currently a Ph.D. candidate in Telematics at the University of Murcia (Spain) being directed by the Ph.D. Pedro M. Ruiz. Nowadays he is working as a Researcher in the area of Telematics Engineering at the Department of Information and

Communications Engineering.

His main research interests are related to ad hoc networks (mobile ad-hoc networks, vehicular ad-hoc networks), modeling efficient solutions for them as well as securing these solutions area within these networks.



security, performance, and scalability.

Daniel Vigueras received the B.Sc. degree in Computer Science in 2009 and a Post-Graduate Diploma in New Technologies in Information and Communications Engineering in 2010 in the Computer Science Faculty of the University of Murcia, Spain. He worked as a Researcher in the area of Telematic Engineering at the Department of Information and Communications Engineering and currently he works as a web architecture designer. His main research interests are about security and efficiency in vehicular ad-hoc networks. His main interests in web applications are related to



(2012), and ADHOC-NOW (2009–2012), and serves as a Reviewer in major IEEE journals and conferences. His main research interests include ad-hoc networking, network performance modeling, distributed algorithms, and new generation networks.

Francisco J. Ros received his B.Sc. degree in Computer Science (2004), Postgraduate Diploma in New Technologies in Information and Communications Engineering (2007), M.S. degree in Advanced Information and Telematic Technologies (2009), and Ph.D. degree in Computer Science (2011) from the University of Murcia, Spain. He is currently working as a Researcher and Part-Time Adjunct Professor for the Department of Information and Communications Engineering (DIIC). He is a Technical Committee Member of IEEE ISWTA (2012), IEEE MASS (2012), Vehi6



Investigator in a number of research projects mainly funded by the European Union, Spanish government, and private companies, and has published a large number of refereed papers in international journals and conferences. He received in 2007 an outstanding research trajectory recognition from the Spanish MEC. He is in the Editorial Board for the Elsevier Computer Communications Journal, the International Journal on Parallel, Emergent, and Distributed Systems, and the International Journal of Network Management. He has served as Chair in the organization of multiple conferences and workshops including among others IEEE INFOCOM, ACM MOBIKOM, ACM MOBIHOC, IEEE MASS, etc. His main research interests include vehicular networks, sensor networks, mobile and ad hoc wireless networks, and distributed systems. He is a Member of the ACM and IEEE Communications Society.

Pedro M. Ruiz received his B.Sc. (1999), M.Sc. (2001), and Ph.D. (2002) degrees in Computer Science from the University of Murcia, Spain. He works as Associate Professor in Telematics at the Department of Information and Communication Engineering (DIIC) at the University of Murcia (UMU). In 2003 he was awarded a *Ramón y Cajal* research position by the Spanish MEC. He has also held Post-Doctoral Research Positions at ICSI in Berkeley, King's College London and University of California at Santa Cruz. During these years he has acted as Principal