Position Verification Approaches for Vehicular Ad Hoc Networks

Tim Leinmüller[+], Elmar Schoch[*], Frank Kargl[*]
[+]DaimlerChrysler AG, Research Vehicle IT and Services, tim.leinmueller@DaimlerChrysler.com
[*]Ulm University, Department of Media Informatics, {elmar.schoch|frank.kargl}@uni-ulm.de

***Abstract*— Inter-vehicle communication is regarded as one of the major applications of mobile ad hoc networks (MANETs). Compared to MANETs, these so called vehicular ad hoc networks (VANETs) have special requirements in terms of node mobility and position-dependent applications, which are well met by geographic routing protocols. Functional research on geographic routing has already reached a considerable level, whereas security aspects have been vastly neglected so far. Since position dissemination is crucial for geographic routing, forged position information has severe impact regarding both performance and security.**

In this work, we first analyze the problems that may arise from falsified position data. Then, in order to lessen these problems, we propose detection mechanisms that are capable of recognizing nodes cheating about their location in position beacons. In contrast to other position verification approaches, our solution does not rely on special hardware or dedicated infrastructure. Evaluation based on simulations shows that our position verification system successfully discloses nodes disseminating false positions and thereby widely prevents attacks using position cheating.

I. Introduction

During the recent years, Mobile Ad hoc Networks (MANETs) have attracted a lot of attention in the research community. Still, there are very few application scenarios where the wide deployment of MANETs is really foreseeable in the near future. One exceptions are networks that inter-connect vehicles on the road, so called Vehicular Ad hoc Networks (VANETs). Main target of research in VANETs is the improvement of vehicle safety by means of inter-vehicle communication. For example in the case of an accident, a VANET might be used to warn approaching cars and give the drivers enough time to come to a halt.

VANETs, especially compared to MANETs, are characterized by several unique aspects. Nodes move with high velocity, resulting in high rates of topology changes, vehicles are equipped with GPS receivers and energy consumption is not an issue. Furthermore, safety applications are time critical and depend on reliable position information. Given these aspects and requirements, geographic routing has been identified to be well suited for VANETs and especially, to perform better than topology-based routing protocols.

An overview on position-based routing schemes for MANETs can be found in [1]. For VANETs, mainly *greedy routing* approaches have been proposed. They have in common that the next hop node of a packet has to be closer to the destination's position than the current node. This implies that a node has to know all its neighbors and their respective position. To achieve that, all nodes send periodic broadcasts of their own position. By this so called *beaconing* every node can build up a neighbor table and base forwarding decisions on it. Two special cases must be handled with greedy forwarding: there might be more than one suitable next hop or there might be no suitable neighbor. *C*ached *G*reedy *Geoc*ast (CGGC) specifically addresses these two cases respecting the special needs of VANETs [2]. In CGGC, the first case as mentioned above is addressed using the neighbor with the minimum Euclidean distance to the target. If no suitable next hop is found, the packet is cached to be forwarded at a later time, utilizing mobility in the network.

While position-based routing protocols like CGGC are very robust under high mobility, there is one critical issue. When nodes send false position information in their beacon messages, this can severely impact the performance of the network. A potential source for such false position data is a malfunction of a node's location sensing system. E.g. a GPS receiver may wrongly calculate the position of a node because of bad reception conditions.

Whereas malfunctioning nodes may degrade the performance of a system to some extent, malicious nodes may cause even more harm. The intents of an adversary may range from simply disturbing the proper operation of the system to intercepting traffic exchanged by ordinary users, followed by a potential modification and retransmission.

In this paper, we first discuss the effects on rout-

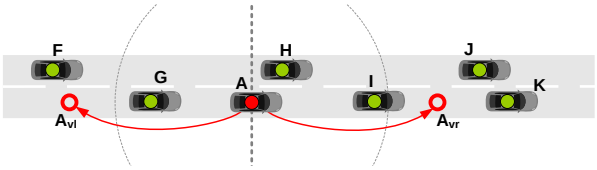

Fig. 1
EXAMPLE SCENARIO

ing arising from false position data. In section III we describe related work on position verification. Then, section IV introduces our verification mechanisms, as well as a framework for the combination of results from different sensors. Section V briefly summarizes the results of our simulative analysis. Finally, we conclude with section VI.

II. EFFECTS OF FALSIFIED POSITION INFORMATION

In this section we outline the influence of false position data generated by malfunctioning or malicious nodes on geographic routing. Figure 1 shows an example scenario where node A claims to be at two additional (faked) positions A_{vl} and A_{vr}. Based on a greedy forwarding strategy, nodes always select the node nearest to the destination as the next forwarding node. Assuming that F wants to send a packet to node K, it will first sent the packet to its only direct neighbor G. G will then forward the packet to the node nearest to the destination from which it received beacons. This seems to be A_{vr}, so the packet ends up at node A, which can now forward, modify or discard it at will. In the opposite direction, the packet from K will go to I, which will again send it to the assumed best node A_{vl}. So faking only two positions, A is able to intercept all traffic along the road.

In order to be able to estimate the impact of falsified position data on geographic routing, we implemented position faking in the ns-2 simulator. For the routing scheme, we use the CGGC approach described in section I, the simulation parameters are summarized in table I. The random waypoint mobility model has been selected to reflect the most general scenario of node movements, e.g. in cities. Malicious nodes are implemented as follows. Whenever a malicious node is about to send a beacon message to announce its present position, it selects a random position on the field and applies it to the beacon (instead of its real position). Whenever a malicious node gets a data packet, depending on the simulation setup, it either forwards it correctly or it drops the packet.

In the following, we present and discuss some of our simulation results regarding the impact of position faking nodes on routing performance. For an in

Parameter	Value
Number of nodes	100
Length of square node field	1000 – 4000m
Node density (nodes/km^2)	6,25 – 100
Max. node velocity (m/s)	50
Mobility model	Random Waypoint
Link-/MAC-Layer	IEEE 802.11
Transmission range (m)	250

TABLE I
SIMULATION PARAMETERS

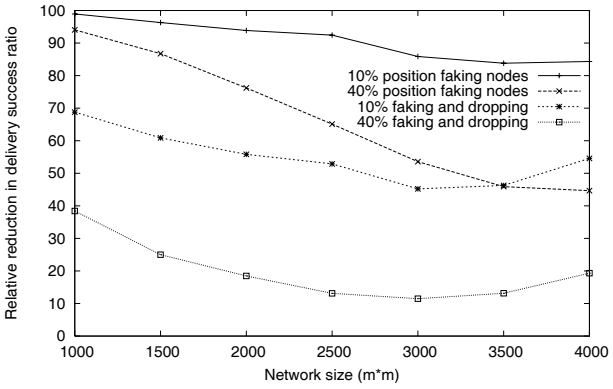

Fig. 2
RELATIVE REDUCTION OF SUCCESSFULLY DELIVERED MESSAGES IN DEPENDENCE OF NETWORK SIZE

detail analysis of routing performance as well as the analysis of reasons for decreased delivery ratio please refer to [3] and [4].

The influence of falsified position information on the overall number of successfully delivered messages has been measured with different percentages of position faking nodes. Figure 2 contains the results of simulation runs with 10% and 40% faking nodes, with and without packet dropping.

The figure clearly shows the detrimental effect of position faking nodes to the overall delivery ratio. If we have 40% position faking nodes which still forward packets, in the worst case this leads to a packet delivery ratio of 50% compared to the delivery ratio without position faking nodes. If position faking nodes additionally drop packets, only about 10% of the packets that would normally reach the destination are actually delivered. We also see that the reduction depends on the simulated network field size. This is the result of two overlapping effects. On the one hand, with increased network size, the number of hops and thus the probability of encountering a malicious node increases. On the other hand, with sparse network density, the probability of unsuccessful delivery due to network partitioning increases anyway and leverages the effects of dropping.

III. Related Work

Whereas a lot of effort was already put in securing traditional MANETs, the security research for position-based routing and VANETs is still in its infancy. Hubaux et al. give an overview on this subject in [5].

The only solution to position falsification attack methods is to introduce some kind of position verification. Some approaches to verify node positions take up the basics of positioning systems. They use angle or distance measurement techniques like radio signal strength or time of flight, partly in combination with challenge-response procedures to securely approve a position claim.

The verification system in [5] relies on base stations building a trustworthy network. In the approach called "Verifiable Multilateration", four of these base stations are involved in every position verification procedure. One after another, each of these stations measures the time between sending a challenge to the corresponding node and the arrival of the answer. Therefore a node might enlarge its actual distance to a base station by delaying the answer, but it is not able to reduce it. In case a node delays the answer and thus enlarges the distance to one base station, this is discovered by a misleading multilateration when looking at all four distance measurements. The approach can be improved by using synchronized base stations. Then, only one challenge message is necessary. The distance can be measured at every involved base station simultaneously. The gain in verification speed is paid with the disadvantage that a node with sectoral antenna can send out the answers to each base station with temporal delay and so is able to trick the verification.

Some other approaches confine themselves to verify that a node resides within a defined region, e.g. for location based access control. The solution in [6] places so called *verifiers* at special locations and defines an acceptable distance for each verifier. Thus a region R can be formed by overlapping circles. The verification procedure then works as follows. First, the corresponding node n sends out a beacon containing its position. Then a verifier v replies with a challenge via radio. After receiving the challenge, n has to answer via ultrasound. If the answer arrives at v in the previously calculated time according to the defined acceptable distance for v, n is approved to be within the region R.

Whereas [6] only works with special hardware, a similar approach in [7] achieves position verification simply based on logic reception of beacons. First, the verifier nodes are divided in acceptors and rejectors. The acceptor nodes are distributed over the region R which is to be controlled. Then, a closed annulus with rejector nodes is formed around the acceptors. In addition to the distinct placement, verifier nodes are synchronized amongst each other. If a node n sends a beacon, the first verifier receiving the beacon decides whether the position of n is acceptable. If the signal first reaches a rejector, n cannot reside within R. If the first reached verifier is an acceptor, n is approved to be in R.

IV. Position Verification Approach

The previously described systems either require specific hardware or rely on an infrastructure of verifiers to check the positions. For many VANET scenarios, these assumptions are not likely to be fulfilled.

In contrast, for VANETs, it would be desirable to be able to verify neighbor position claims without any additional or dedicated devices. Therefore, in our solution, we use the concept of a "Position Cheating Detection System" similar to intrusion detection systems to detect e.g. selfish nodes in MANETs [8]. In these systems each node uses multiple sensors to detect malicious or selfish behavior of nodes in the network. Based on the sensors' observations, each node calculates a trust value that determines whether nodes are trustworthy or should e.g. be excluded from further routing decisions. Such a system can predict the trustworthiness of other nodes even when single sensors do not work reliably to hundred percent.

We now transfer this idea to the domain of position verification. Therefore it is only necessary to find suitable sensors that can be used to detect cheated position information. Basically, there are two classes of position verification sensors. Sensors of the first kind work autonomously on each node and contribute their results to the overall trust ratings of neighbors. The second class includes sensors that only work in cooperation with other nodes surrounding the neighbor node in question.

All sensors suggested have the benefit that they only rely on information that the routing layer delivers anyway, so there is no extra hardware involved. Additionally, only the normal nodes forming the VANET are included. So there is also no need for a dedicated infrastructure.

A. Combination of Verification Sensors

The accumulation of observations over time and sensors is required to provide a decision, whether a node is to be regarded as being malicious or not. Also knowing that observations from some sensors are more reliable than observations from other ones, we use a trust model that provides the capabilities to

consider observations from differently weighted sensors during a certain period of time. The mathematical model mainly derives from the one presented in [9].

When we denote the n-th observation of sensor s by σ_n^s, the trust model can be described as follows:

- All nodes store trust values $r \in [-1; 1]$ for all direct neighbors. $r = 0$ is equivalent to neutral trust, $r \in (0; 1]$ means a node is trustworthy and $r \in [-1; 0)$ means no trust.
- Every observation σ_n^s is stored with timestamp t_n^s.
- On the arrival of a new observation, the trust value for a neighboring node is recalculated according to the collected observations for this node.
- All observations are stored for a maximum time T and discarded afterwards.

The weight factor w^s of an observation σ_n^s is chosen according to the reliability of the providing sensor, e.g. observations from a more reliable sensor like ART can be regarded as more valuable than observations from a less reliable one like MGT sensor (see next section for description of sensors). Besides, observations may also be weighted dynamically, e.g. if a sensor delivers observations different reliability each.

The timestamp t_n^s of an observation σ_n^s is used to calculate the observation's time factor $wt(t, t_n^s)$,

$$wt(t, t_n^s) = 1 - \left(\frac{t - t_n^s}{T}\right)^x$$

where x denotes the exponential aging factor of the observations. $x = 1$ corresponds to a linear aging process, values $x > 1$ are equivalent to a more than linear aging process of the respective observation.

Finally, the trust value r_t of a neighbor node at a time t is calculated by multiplying the available observations by their weight factor and their time factor, then summarizing the results and at the end normalizing to $[-1; 1]$.

Detected violations are weighted higher than observations of normal behavior, thus once a falsified position information is detected, it takes several correct beacon messages to compensate the trust level.

In the routing protocol, location information is distributed between nodes by means of position beacons. In order to prevent abuse of the verification system, beacons need to be authenticated and timestamped by their sender. When a node receives a position beacon from another node, claiming to be at a certain position, the sensors get active to verify if this claim is likely to be correct or not.

Next we present different sensors that can be used in our architecture. The first class of sensors works autonomously, whereas the second class needs a set of cooperating sensors.

B. Autonomous Sensors

1) Acceptance Range Threshold: The Acceptance Range Threshold (ART) sensor is based on the observation that all radio networks have a maximum communication range where packets sent by a node B can still be received successfully by a node A. Based on the radio used in VANETs, we define a maximum acceptance range threshold Δ_{max}.

By discarding position beacons from nodes claiming to be at a distance larger than Δ_{max} away from a receiving nodes' current position, we avoid many types of attacks. Using this simple method, nodes e.g. cannot easily collect all outgoing traffic of a node by pretending to be at a better forwarding position than potential other nodes nearby.

2) Mobility Grade Threshold: The Mobility Grade Threshold (MGT) is based on the assumption that nodes can move only at a well-defined maximum speed. Depending on the scenario, this may be the general speed-limit on streets (plus a bonus for speeding cars) or the maximum walking speed of persons. When receiving a beacon, nodes also record a timestamp. Upon the reception of subsequent beacons from the same node, it is checked whether the average speed of the node between two positions in the two beacons exceeds the MGT. If yes, the beacons are discarded.

Whereas this sensor detects rapid changes in a node's alleged position, it cannot detect gradual changes where nodes slowly change their position claim towards a wrong direction.

3) Maximum Density Threshold: Similar to the last sensor, this sensor is based on the assumption that only a restricted number of physical entities (e.g. cars) can reside in a certain area. For instance, cars have certain physical dimensions preventing too many cars to be on the same road segment. This sensor defines a Maximum Density Threshold (MDT) which, when exceeded, rejects further position beacons for this area. It aims at preventing so called Sybil attacks, where a node creates a large number of virtual nodes in order to collect all traffic in a certain area.

4) Map-based Verification: Here we assume that many cars include car navigation systems where street maps are accessible by the position verification system. Then the system can check whether cars pretend to be at impossible locations, e.g. off the streets, in houses, etc.

5) Overhearing: Overhearing is a concept introduced by Marti et al. [10] where nodes use the so-called promiscuous mode to capture packets that are

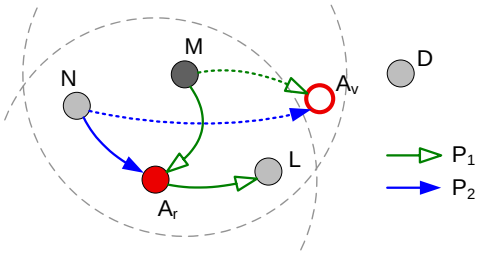

Fig. 3
OVERHEARING

sent by nodes in reception range but are addressed to other nodes. Whereas Marti et al. use this concept to control forwarding behavior of nodes, we use it to verify position information. As shown in figure 3, there are two cases where overhearing is useful (Note, A_r represents the real position of node A, whereas A_v denotes the position, A pretends to be by sending it to neighboring nodes in its beacon messages).

In first case node M forwards packet P_1 to node A. Later M overhears P_1 being sent to node L which is at a inferior position (with regard to the routing metric) compared to A. This indicates that A may have forged his position A_v. In the second case node M overhears the transmission of packet P_2 from N to A, although given the last position of A known to M and the Mobility Grade Threshold, A should not be in reach of N. Again this indicates that A may have forged his position A_v.

Whereas the earlier sensors are quite reliable, the overhearing sensor gives only indications that position information may have been forged. But there are valid cases where the sensor will wrongly detect nodes to spoof positions. So the overhearing sensor might only be used as trigger to take additional actions like the ones described in the next section.

C. Cooperative Sensors

In contrast to autonomous sensors, cooperative ones need to communicate and exchange information in order to detect position faking nodes. Whereas this creates some overhead, it also offers the opportunity to further increase the detection rate compared to pure autonomous sensors. In order to reduce this overhead, such sensors may be used only when autonomous sensors indicate that some position faking may be going on.

It is important to note that these mechanisms, if not secured properly, may also create additional attack opportunities. Position faking nodes may send wrong information messages to its neighborhood or extract information from the exchanged packets to even improve their position faking.

1) Proactive Exchange of Neighbor Tables: Here nodes exchange their neighbor tables and then check if the positions received correspond to their own data. One can further distinguish whether the exchanged neighbor tables include the position of neighbors or only the fact that two nodes share a common link.

In the first case, when a node A receives a beacon from node B claiming to be at position P_B and receives a neighbor table from node C containing the information that B is at position $P_{B'}$ and P_B and $P_{B'}$ differ significantly, A can conclude that one of the position claims must be false. In this case it cannot determine if B is sending false information or whether C has modified the information in its neighbor table. When more neighbors distribute their neighbor tables, A can take a majority decision whether to believe the position claim of B or not.

If C sends the neighbor table without position information, A can apply a verification mechanisms similar to the ART sensor to check if B at its claimed position P_B is in the range of C (then it must appear in the neighbor table of C) or if B is outside the transmission range of C (in this case B must not appear in the neighbor table of C).

Of course these checks have only statistical significance and thresholds must be applied to prevent too many false-positives. Further the results are not taken directly as a base for the decision which position information to drop, but are combined with other observations as described in section IV-A. So only the combination of multiple observations lead to the rejection of a position claim.

2) Reactive Position Requests: For this sensor, nodes only cooperate for position verification upon demand. This could be triggered when a node A encounters an other node B which it has never met before. Besides, demand to verify position claims of an already known neighbor could be raised by indications from autonomous sensors that the node has started to cheat about his position.

Thus, the corresponding node A starts the verification process by selecting several neighbors as acceptor or as rejector. Because A knows the positions of its own neighbors, the claimed position of B as well as the theoretical transmission range of the radio hardware, A is able to distinguish between own neighbors that should have received beacon messages of B and those that should have missed beacons because their distance to B is too large. Hence, it randomly selects some rejectors among those neighbors that should not have received a beacon from B and some acceptors from those that are supposed to have received one. Having recorded this, A sends out a position request (PREQ) in which all acceptors and rejectors are asked

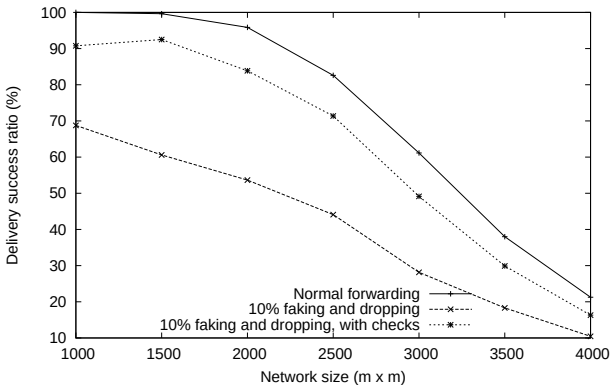

Fig. 4
DELIVERY SUCCESS RATIO WHEN USING THE VERIFICATION SYSTEM

for the position of B. In case an addressed node does not know B yet, it needs to answer as well with an according message.

After received the responses, A is able to compare them with what it expected and can rate the position claim of B. For instance, the more supposed rejectors actually got the current beacon from B, the lower the sensor output will be, because this indicates that B has given a falsified position.

V. Position Verification Analysis

As evaluation of the presented verification techniques, we added the mechanisms to the simulation environment as described in section II.

From the simulation results in figure 4 we can see that compared to a system without position verification, detecting malicious nodes and excluding them from routing results in improved delivery ratios. We found that our framework was able to detect position faking nodes with an accuracy of 95% in most scenarios.

VI. Conclusions

Falsified position information in mobile ad hoc networks with geographic routing protocols results in network performance degradation and allows attackers to intercept packets. In this work we have analyzed the effects of falsified position information in VANETs. Our simulation results show that the overall delivery ratio might decrease significantly. Whereas for scenarios without packet dropping by position faking nodes, drops resulting from routing loops are the main reason, in scenarios with packet dropping the dropping itself is the main cause.

As a countermeasure, we have presented a framework to detect and mitigate the influence of falsified position information in geographic routing protocols. In contrast to other position verification approaches, we do not rely on special hardware nor on preinstalled infrastructure. Our goal is to quickly estimate the trustworthiness of the position claims of neighbored nodes.

The selected mechanisms will not entirely prevent malicious nodes from using falsified position information, however, they will significantly limit the options of position faking nodes (i.e. fake positions must meet all criteria as opposed by the deployed sensors, for instance they must reside within a node's wireless transmission range). We have shortly outlined simulation results showing the effectiveness of our approach. Future work will enhance the simulation scenarios and implement more sensors.

References

[1] Holger Füssler, Martin Mauve, Hannes Hartenstein, Michael Käsemann, and Dieter Vollmer, "A Comparison of Routing Strategies for Vehicular Ad Hoc Networks," Technical Report TR-3-2002, Department of Computer Science, University of Mannheim, July 2002.

[2] Christian Maihöfer, Reinhold Eberhardt, and Elmar Schoch, "CGGC: Cached Greedy Geocast," in *Proceedings of 2nd Intl. Conference Wired/Wireless Internet Communications (WWIC 2004)*, Frankfurt (Oder), Germany, Feb. 2004, vol. 2957 of *Lecture Notes in Computer Science*, Springer Verlag.

[3] Tim Leinmüller, Elmar Schoch, Frank Kargl, and Christian Maihöfer, "Influence of Falsified Position Data on Geographic Ad-Hoc Routing," in *Proceedings of the second European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS 2005)*, July 2005.

[4] Tim Leinmüller and Elmar Schoch, "Greedy routing in highway scenarios: The impact of position faking nodes," in *Proceedings of Workshop On Intelligent Transportation (WIT 2006)*, Mar. 2006.

[5] Jean-Pierre Hubaux, Srdjan Čapkun, and Jun Luo, "The Security and Privacy of Smart Vehicles," *IEEE Security and Privacy*, vol. 4, no. 3, pp. 49–55, 2004.

[6] Naveen Sastry, Umesh Shankar, and David Wagner, "Secure verification of location claims," in *Proceedings of the 2003 ACM workshop on Wireless security (WiSe'03)*. 2003, pp. 1–10, ACM Press.

[7] Adnan Vora and Mikhail Nesterenko, "Secure location verification using radio broadcast," in *Proceedings of 8th International Conference on Principles of Distributed Systems (OPODIS 2004)*. 2004, Springer Verlag.

[8] Frank Kargl, Andreas Klenk, Stefan Schlott, and Michael Weber, "Advanced Detection of Selfish or Malicious Nodes in Ad Hoc Networks," in *Proceedings of 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*. Sept. 2004, pp. 152–165, Springer Verlag.

[9] Pietro Michiardi and Refik Molva, "CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, Deventer, The Netherlands, The Netherlands, 2002, pp. 107–121, Kluwer, B.V.

[10] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 255–265.