Securing Multihop Vehicular Message Broadcast using Trust Sensors

Matthias Gerlach, Oleksandr Mylyy, Nestor Mariyasagayam, and Massimiliano Lenardi

Abstract The ad hoc wireless exchange of position and velocity information between vehicles enables a plethora of new applications that can increase the safety and efficiency of driving. Efficient and reliable flooding mechanisms for vehicular applications mandate correct and timely received positions, vehicular safety applications even more so. This work first assesses the impact of different position faking attackers on the "goodput" of Multi-Hop Vehicular Beacon Broadcast (MHVB-B), a dissemination mechanism for vehicular networks. Then we use a set of known and simple heuristics to improve the detection of fake positions within MHVB-B data and briefly assess their impact on the goodput. At the core of this work, we define a framework for integrating arbitrary trust sensors using Bayesian reasoning and describe a way to determine their contribution to the overall assessment of message trustworthiness, that we model as a conditional probability.

1 Introduction

Vehicular communication based on wireless short-range technology facilitates a plethora of new applications at low cost for safety, traffic efficiency, and infotainment using direct or multi-hop communication. Securing vehicular communication does not only involve *preventive security measures* to ensure access control (authorization) of nodes and ensure integrity, authenticity, and confidentiality of messages. It also involves *detection and reaction methods*. This is particularly true in large networks, where integrity and authenticity

Matthias Gerlach (e-mail: matthias.gerlach@fokus.fraunhofer.de) and Oleksandr Mylyy are with Fraunhofer Institute for Open Communication Systems (FOKUS). Nestor Mariyasagayam and Massimiliano Lenardi are with Hitachi Ltd. Sophia Antipolis Labs (HSAL).

Please use the following format when citing this chapter:

Gerlach, M., Mylyy, O., Mariyasagayam, N., Lenardi, M., 2008, in IFIP International Federation for Information Processing, Volume 265, Advances in Ad Hoc Networking, eds. Cuenca, P., Guerrero C., Puigjaner, R., Serra, B., (Boston: Springer), pp. 109-120.

do not guarantee the correctness of transmitted data due to insider attacks or faulty nodes.

For many vehicular applications, *mobility data beaconing* is necessary for cooperative message distribution (*routing*) or cooperative safety applications (*cooperative awareness*). Each node periodically sends out mobility data, i.e., position and velocity. The algorithms for mobility data beaconing have to introduce little overhead, and be scalable and robust in order to account for the properties of vehicular networks, i.e., short vehicle contact times, the network size, and the adverse radio environment. Consequently, important *quality of service parameters* for mobility data beaconing are timeliness of delivery (success rate) and how much of intended geographic area really has been covered. Mariyasagayam et al. present an algorithm for vehicular beaconing in pre-defined geo-regions by means of efficient flooding in [1] and an enhanced version in [2]. The algorithm, called MHVB-B, Multi-Hop Vehicular Beacon Broadcast, achieves a higher message dissemination success rate by selectively repeating messages based on distance from the originator. In this paper, we will discuss and assess the impact of position faking attacks on MHVB-B.

This paper is organized as follows. In Section 2 we discuss the objectives an attacker pursues with mobility data manipulation. Section 3 presents simulations we carried out on the impact of position faking nodes on MHVB-B and the impact of simple checks to assess incoming positions. In Section 4 we outline a novel trust framework and describe how arbitrary trust sensors can be integrated into the framework. Section 5 concludes this paper and outlines future work.

1.1 Related Work

With respect to security, it is commonly agreed that anonymous certificates and digital signatures should be part of a security solution for vehicular communication. IEEE 1609.2 [3], for example is a standard describing how to secure messages for vehicular communication using ECDSA as a mandatory asymmetric crypto-algorithm. Solutions based on certificates have also been proposed by Raya et al. [4] and Festag et al.[5], and Gerlach [6], for example. For the sake of this work, we assume certificates and signatures to prevent Sybil attacks. In [7], Golle et al. present a general approach to assess the validity of data in VANETs. Based on available sensor information, each node builds a model and attempts to find the simplest explanation for the received data with this model. The paper focuses on presenting a theoretic framework. Leinmüller et al. carried out simulations to evaluate the impact of position-faking nodes on geographic forwarding [8] and propose position verification algorithms in [9]. In particular, they found that using plausibility checks can detect the majority of false position reports.

2 Mobility Data Manipulation

Mobility data faking describes the process of changing a node's true position and velocity to fake values with the purpose of injecting those into the vehicular network. Current work identifies position faking or manipulation as a probable attack and describes ways to inject fake data into the network, e.g., by manipulation of sensor input to the on board unit. Aijaz et al. describe ways to manipulate the input to the on board unit [10, Figure 6], and hence the way to achieve the attack. Assuming the attacker has found a way to inject false movement information into the network, it should be discussed what kinds of change to mobility data can be detected – and what their potential impact may be.

Faking its identity, a node carries out a *Sybil* attack – potentially with different sets of mobility data. A Sybil attack can make the recipients believe that there is a traffic jam ahead, for example. For the sake of this work, we assume that Sybil attacks can be prevented using short lived certificates for vehicle authorization as discussed in Section 1.1. This reduces our discussion to faking position, speed and heading.

Faking the position of a node means that an offset is added to the position before it is sent out such that the fake position matches the objectives of the attacker. The attacker's objective can be to attack a certain node, or a certain mechanism, independent from the node. The same holds true for velocities, i.e., heading and speed. Consequently, we need to discuss the attacker's objectives in order to find out probable attacks with changed positions and velocities.

For vehicular settings, geo-addressing, i.e., rather addressing a geographic region than a certain node, is an important feature. Typically, there are two different phases for geo-addressed delivery of packets: *line-forwarding* and *area-forwarding*. The first is used to efficiently transport the message to the area of interest and the latter is used for disseminating the information within the target area. Position based routing mechanisms are often used for the line-forwarding. For area-forwarding, flooding is an example for a simple strategy. While attacks on greedy forwarding for line forwarding have been looked at by Leinmüller et al. (cf. Section 1), we will discuss the impact of position faking nodes at MHVB-B, i.e., an efficient area forwarding algorithm.

The objective of an attacker with respect to the networking mechanism will be the disruption of communication or the isolation of certain nodes. Looking at the above requirements, an attacker will attempt to spoil timeliness of delivery, and to decrease the number of nodes that can be reached within the destination region. The only way to achieve this is by faking movement information that prevent *all* potential forwarders sending their message to the next hop. Potential forwarders in MHVB-B are selected implicitly and in a distributed fashion as a function of distance to the sender. By overhearing the channel, MHVB-B nodes avoid duplicate retransmissions after receiving a packet from a node with a larger distance to the sender than itself. Conse-

quently, a potentially viable attack on MHVB-B would be a forwarder that fakes its own position to be farther away from the source node (and calculates its back-off accordingly). Like this, receiving nodes in optimal position for retransmission would back-off and the packet may not reach all intended receivers in the defined region. Apart from this, the most relevant threat will be the dissemination of false positions for the use in applications, such that the attacker can cause accidents, gain an advantage (a free road) or otherwise use the system for harming people.

3 Simulation Results – Attacker Impact

For the analysis of possible attacks on mobility data we created a simple attacker framework in ns-2. We use it to test the influence of different attacks on MHVB-B. The malicious behavior of an attacker node can be realized through the assignment of an attack function, respective parameters, and attack timing to this node. The attack function parameters contain the type of attack that should be carried out. From the implementation side, the attacker code can easily be integrated into existing mobility aware nodes by inserting a *manipulate()-function* into a simulation agent's code before submitting positions for sending. Configuration of the attacker is done in a dedicated file using tcl according to the typical ns-2 configuration files.

Attacker Type	Used Parameters	Description
Normal node	attacker_functype	The node works as expected
OFFSET	Attacker *adds a predefined offset* to the real position	
	attacker_xpos_dist, attacker_ypos_dist	Value to define the value of the offset to be added
	attacker_random_xpos, attacker_random_ypos	Flag to define which direction shall be changed by attacker (x, y, or both). Value is changed within a predefined range

Table 1 Attacker framework configuration parameters for OFFSET attacker

For the purpose of this work we consider the offset type as the most commonly used attacker type. Table 1 depicts the major parameters for the attacker. If the offset value is calculated not in a random manner but according to the current goal, these attackers may be widely used for modeling different situations. The offset attacker may sensibly model a malicious node with a purpose.

3.1 Impact on MHVB-B

We run simulations for MHVB-B using our attacker framework with a variety of attackers. To estimate the effectiveness of the MHVB-B, the performance parameter success rate (sr) was defined for a node as the following ratio [1]:

$$sr = \frac{pkt\_number\_rcvd\_within\_threshold}{total\_rcvd\_pkt\_number}, where \quad (1)$$

pkt_number_rcvd_within_threshold is the number of packets received by a node within the threshold=0.3s and is within the 400m radius of the originator, and *total_rcvd_pkt_number* is the total number of packets received by the same node during the entire simulation time. sr did not change significantly due to the definition of sr: if an intermediate attacker node drops received packets, the receiver node does not know that these packets were issued by the sender node and, as a result, does not change its *total_rcvd_pkt_number*. By contrast, for simulating the impact of position faking nodes on MHVB-B, packets lost inside the network due to the different malicious actions, dropping packets etc. must additionally be taken into account by the performance analysis algorithm. Therefore, in order to make the attacker impact on MHVB-B visible, we define a new performance parameter, called *message goodput*. This parameter can be described as follows:

$$message\_goodput = \frac{good\_pkt\_number\_rcvd\_within\_threshold}{total\_rcvd\_pkt\_number} \quad (2)$$

where *good_pkt_number_rcvd_within_threshold* is the number of packets with correct, i.e., usable, movement information received by a node within the threshold=0.3 seconds and is within the 400m range of the originator, and *total_rcvd_pkt_number* is the total number of packets received by the same node during the entire simulation time. For our simulation, we counted correct and detected false positions as contribution to the good packet number. We argue that even some of the detected false positions may be used, depending on the given and assumed accuracy.

Figure 1 depicts the impact of position manipulation attacks on the goodput of MHVB-B for different attackers and attacker densities, and different movement scenarios as a function of distance between nodes. As first movement scenario we chose a single lane scenario with the length of 10 km and the node density of 30 node per km similar to the one described in [1]. For the second scenario, we took a realistic highway movement scenario with two lanes 15 km and the average node density of 7 nodes per km. Simulation time was 2 min for both cases. All attackers were offset attackers with different offset values. Together with the normal mode (without attacker in the network) we studied the attacker penetration of 20%, 40%, and 80% for each scenario. Figure 1 shows that goodput is degraded by the attacker by about

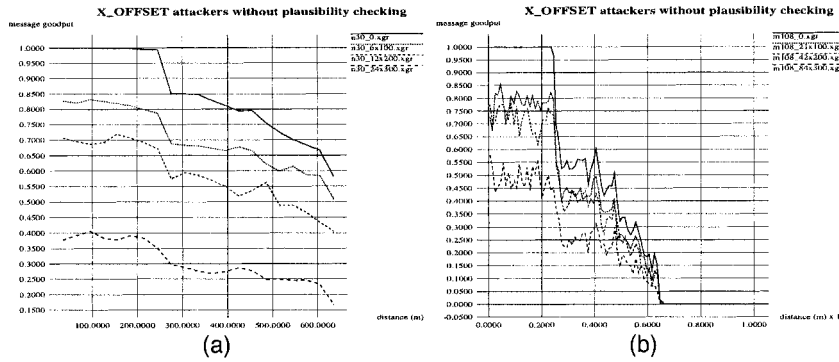

Fig. 1 Impact of different attackers on MHVB-B goodput without plausibility control. (a) single lane scenario (b) highway scenario

the amount of attackers in the network for each distance. We can observe a significant difference between the normal case and the offset attackers (with the offset value 300m and the penetration rate of 80%) over the entire simulation time with no significant difference between the two scenarios in terms of goodput degradation.

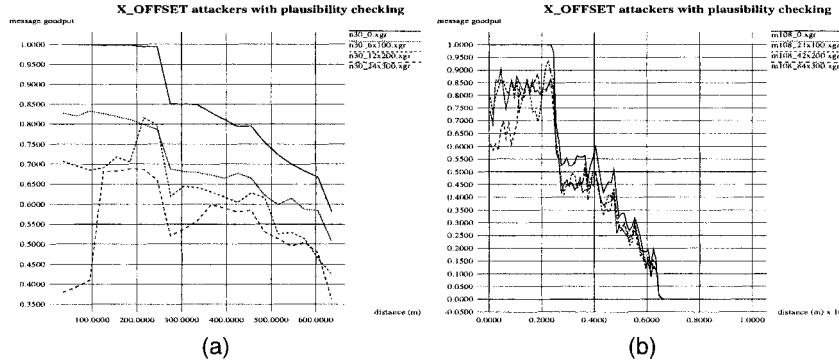

Fig. 2 Using plausibility mechanisms to improve goodput. (a) single lane scenario (b) highway scenario

Figure 2 presents the simulation results after applying simple plausibility control mechanism which comprised the plausibility checking using two well-known plausibility sensors – ART (acceptance range threshold) and MGT (mobility grade threshold) (cf. [9]). ART accepts positions only within a certain range; messages containing a position further away than, e.g., 500m

are assumed to contain a false position. MGT assumes that the delta between two positions can only be such that a realistic velocity is not exceeded.

The impact of the plausibility checking can be derived from comparing this figure with Figure 1. We observe a significant improvement in message goodput for all cases. This fact, as well as the remaining difference in message goodput between the normal mode and the integrated attackers, is due to the relatively weak plausibility checks, which only detect significant position deviations and calls for the integration of more elaborate mechanisms. We are currently developing those for the use of our framework based on the Kalman filter.

4 Detecting and Handling False Mobility Data

The discussions and results in the previous sections identified the need for the assessment of movement information and appropriate measures of reaction and more elaborate checks. In this section, we will discuss a probabilistic framework detection of false movement data.

At the core of our discussion are the notions of trust and trustworthiness. While trust typically contains the element of a decision already – we decide to *trust* somebody or not – this is typically based on the evaluation of the *trustworthiness* of that person, its statements and behaviour. Along those lines, an application will decide to trust the given data, while our system attempts to assess the trustworthiness of this data beforehand. This distinction should be borne in mind for our trust sensors, which are really trustworthiness sensors.

4.1 Detection and Fusion – Trust Evaluation

Taking up the ideas of *security sensor fusion* of Gerlach et al. [11] and refining their work, we are currently implementing a framework for detecting and tagging movement information based on the input from different trust sensors. Incoming mobility data messages are assessed in the trust evaluation module that uses Bayesian inference for obtaining a statement about the trustworthiness of the given data using a value in the interval $[0, 1]$ that can be interpreted as a probability. The trust decision is then taken by mobility data users. Both trust evaluation and decision methods may use context data to include more information about the environment for more accurate decisions.

Figure 3 depicts the input, output and data fusion process of our trust framework. A simple set of movement information comprises a node's position, speed, and heading. Acceleration would complete the picture, but we do not take it into account here. Note that a set of movement information is

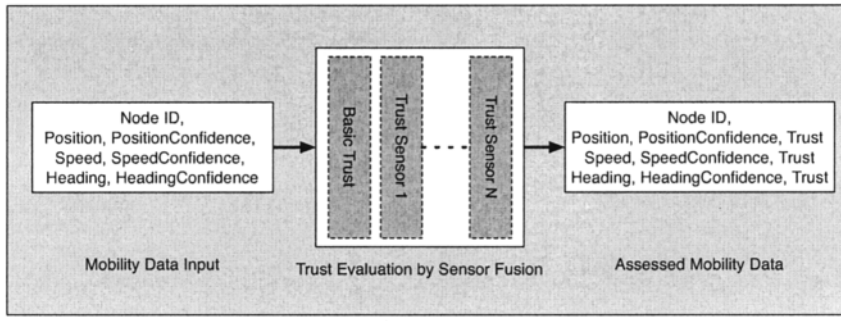

Fig. 3 Data flow in the sensor fusion framework

atomic, in the sense that it must be evaluated as a whole. For our trust evaluation system we assume that (1) accuracy information (confidence values) is an integral part of the movement information, (2) trust sensor information is fused using recursive application of Bayesian inference rule (the output of the n-th sensor is the input to the $n+1$-th reasoning step), and (3) we can estimate the first prior probability as the basic trust of the system as discussed in Section 4.2.1.

At the core of Bayesian inference – a well-known method of statistical inference – is the inversion formula $P(C|sensor) = \frac{P(sensor|C)P(C)}{P(sensor)}$. In terms of probability ratios, the formula can be written as (see also [12]):

$$\frac{P(C|sensor)}{P(\neg C|sensor)} = \frac{P(sensor|C)}{P(sensor|\neg C)} \frac{P(C)}{P(\neg C)} \tag{3}$$

Taking positions as an example, the hypothesis C translates to "the position is correct". *sensor* is the notation to describe the knowledge and corresponding algorithms used in the sensor. The probability $P(C|sensor)$ describes the probability C is true given the sensor reading. This translates to the *trustworthiness* of the position given the sensor reading. The ratios in Equation 3 are typically described as *odds*, *likelihood ratio*, and *prior odds*. The relevant input from a trust sensor is the likelihood ratio. Given the odds after one sensor's likelihood ratio contribution, another's likelihood ratio can be used to recursively update trustworthiness in Equation 3.

4.2 Trustworthiness Sensors

Every trust(worthiness) sensor provides the likelihood ratio $L(sensor|C)$ as input to the fusion process:

$$L(sensor|C) = \frac{P(sensor|C)}{P(sensor|\neg C)} \tag{4}$$

Equation 4 requires us to estimate the probabilities for $P(sensor|C)$, i.e., the probability that our sensor confirms a correct message, and the probability that our sensor falsely confirms a wrong message ($P(sensor|\neg C)$). Sometimes we can directly estimate this ratio not even using probabilities, sometimes we need to use repeated trials (or a virtual experiment) to obtain these values. Note that it is not necessary that these two probabilities sum to unity.

4.2.1 Basic Trust Sensor – Prior Trust

The choice of the first prior probability, or equivalently the prior odds is crucial for using the Bayes formula in the recursive form. We propose to interpret the prior probability as the basic trust a node has in its environment. Basic trust models the general trustworthiness a node assigns to a statement before even assessing it, and is typically a function of a node's prior experiences.

For our communication based system this probability represents our belief that a message is correct when we receive one. Again given our Hypothesis C from above, this can be written as conditional probability $P(C|basic)$, where *basic* is the statement "we received a message". Assumed, due to system measurements, an in depth risk analysis, and appropriate security measures, we knew the average fraction of correct messages received is 99% then we could directly set this value as basic trust. For future definitions of basic trust our framework also allows the definition of individual, time, and position dependent basic trust.

4.2.2 Reception Range Threshold

Leinmüller et al. defined the acceptance range threshold (ART) sensor as a simple heuristic that only accepts positions from within the acceptance range [9]. The rationale behind ART is that it is impossible to receive a message from beyond the reception range due to the radio system constraints. Even though intuitively correct, it is difficult to model the power of trustworthiness statements and hence its contribution to the overall assessment to our sensor fusion system. Consequently, we extend the model of the acceptance range to what we call the *reception-range-sensor* to make it more accurate, and be able to include it into our sensor fusion framework.

Figure 4 depicts the underlying model for the reception range sensor, both how it can be visualized in real life (Figure 4-(a)), as well as a function of the reception probability over the distance (Figure 4-(b)). $P(d)$ could be an arbitrary function modelling the relation between reception probability and distance to the sender. Now defining d_{NodeB} as the claimed distance of Node

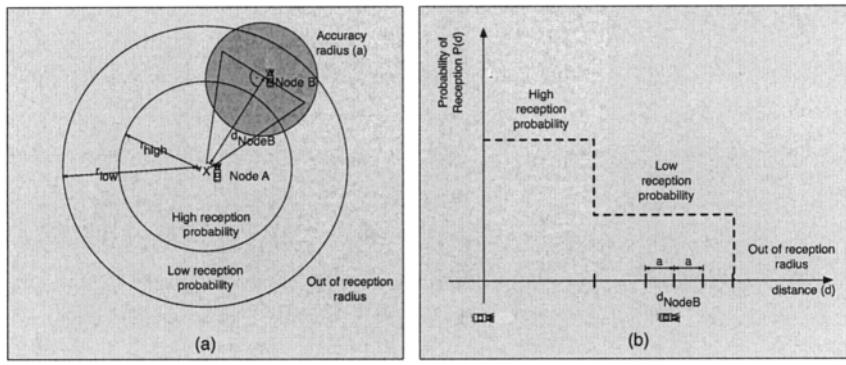

Fig. 4 Model of the reception probability sensor: (a) Overview (b) Probability of reception function over distance

B from the receiving node, a as the accuracy of this claim, we can calculate the likelihood ratio for this sensor as follows:

$$L_{dist} = \frac{\int_{d_{NodeB}-a}^{d_{NodeB}+a} P(\delta)d\delta}{\int_{0}^{\infty} P(\delta)d\delta - \int_{d_{NodeB}-a}^{d_{NodeB}+a} P(\delta)d\delta} \tag{5}$$

Note that the transition from reality to the model depicted in Figure 4 does not account for the circular area around the position of Node B, to reduce complexity of the algorithm. Similarly, we do not take into account the accuracy of node A's position as we assume that it knows its position with a relatively high accuracy. Further, the above likelihood ratio only takes into account the distance from the receiver. Hence, it is not yet complete: the position can still be at any angle from the distance. Therefore, we need to add the probability that the given position is indeed within the direction it claims to be. The likelihood ratio can be calculated as:

$$L_{angle} = \frac{\Psi}{\pi} = \frac{\arctan \frac{a}{d_{NodeB}}}{\pi} \tag{6}$$

Ψ describes the angle spanned by the right angle triangle a (opposite leg) and d_{NodeB} (adjacent leg) as depicted in Figure 4-(a). As the distance and the angle are independent by the definition of polar coordinates, we can combine the two likelihood ratios to obtain the new likelihood ratio:

$$\frac{P(RRS|H)}{P(RRS|\neg H)} = L_{RRS} = L_{distance} L_{angle} \tag{7}$$

L_{RRS} is the overall likelihood of our reception range sensor counting the evidence RRS. We can now obtain $P(sensor|C)$ and $P(sensor|\neg C)$ by requiring the numerator N and denomitator D of Equation 7 to sum to unity.

4.3 Trust Decision – Handling False Data

In general, every application or networking mechanism defines its own way to deal with untrustworthy data. It can define the level below which it deems something too untrustworthy to act upon or it acts in the case of untrustworthy information – such as an intrusion detection system. In addition, this trust threshold may also depend on the situation the given node is in.

In the case of MHVB-B, different choices are possible. In first simulations carried out in this work, an MHVB-B node receiving false information dropped that particular message. As a result any (detected) false message was eliminated in the first hop. As repeaters may also cheat with their position and thus have an adverse impact on the reachable geographic region it may be prudent for intermediate nodes to check the position of the forwarder and – if this is not trustworthy – resend the packet. Even though this may result in a higher channel use, it would also make the protocol more resistant against false positions.

5 Conclusion and Future Work

In this paper, we evaluated the impact of position faking attackers on MHVB-B, an efficient dissemination protocol. After the initial results presented here, we conclude that the networking mechanism of MHVB-B seems to be fairly robust to position faking. For better visualization of the impact of position faking nodes, we created a new measure – goodput – that reflects the number of usable positions sent around in the network. We used well-known and simple plausibility mechanisms to improve the goodput and conclude that a more elaborate framework with more accurate trust sensors must be developed that we presented thereafter.

Future work will include to improve our attacker framework and carry out more simulations confirming the robustness of MHVB-B. Further we need to find a more accurate metric to reflect the impact of position faking to the networking mechanism. Last but not least, we will implement, evaluate and use our sensor-fusion framework to integrate more trust sensors, and develop more accurate trust sensors.

References

1. Mariyasagayam, N., Lenardi, M.: Broadcast algorithms for Active Safety Applications over Vehicular Ad-hoc Networks. In: Proceedings of 4th International Workshop on Intelligent Transportation (WIT), Hamburg, Germany (2007)
2. Mariyasagayam, N., Osafune, T., Lenardi, M.: Enhanced Multi-Hop Vehicular Broadcast (MHVB) for Active Safety Applications. In: Proceedings of 7th International Conference on ITS Telecommunications (ITST). (2007)
3. Institute of Electrical and Electronics Engineers: IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages (2006)
4. Raya, M., Papadimitratos, P., Hubaux, J.P.: Securing Vehicular Communications. IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications (2006)
5. Fonseca, E., Festag, A., Baldessari, R., Aguiar, R.: Support of Anonymity in VANETs – Putting Pseudonymity into Practice. In: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong (2007)
6. Gerlach, M., Rechner, H., Leinmüller, T.: Security Framework for Vehicular Applications. In Altintas, O., Chen, W., eds.: Proceedings of Third International Workshop on Vehicle-to-Vehicle Communications (V2VCOM), Istanbul, Turkey (2007)
7. Golle, P., Greene, D., Staddon, J.: Detecting and Correcting malicious data in VANETs. In: Proceedings of the first ACM workshop on Vehicular ad hoc networks (VANET). (2004) 29–37
8. Leinmüller, T., Schoch, E.: Greedy Routing in Highway Scenarios: The Impact of Position Faking Nodes. In: Proceedings of Workshop on Intelligent Transportation Systems (WIT 2006). (2006)
9. Leinmüller, T., Schoch, E., Kargl, F.: Position Verification Approaches for Vehicular Ad Hoc Networks. IEEE Wireless Communications Magazine (2006)
10. Aijaz, A., Bochow, B., Dötzer, F., Festag, A., Gerlach, M., Kroh, R., Leinmüller, T.: Attacks on Inter Vehicle Communication Systems - an Analysis. In: Proceedings of Second Workshop on Intelligent Transportation Systems (WIT). (2005)
11. Gerlach, M., Festag, A., Leinmüller, T., Goldacker, G., Harsch, C.: Security Architecture for Vehicular Communication. In: Proceedings of Fourth Workshop on Intelligent Transportation Systems (WIT), Hamburg, Germany (2007)
12. Pearl, J.: Probabilistic Reasoning – Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc., San Mateo, California (1993)