Design and Performance of Secure Geocast for Vehicular Communication

Andreas Festag, Senior Member, IEEE, Panagiotis Papadimitratos, Member, IEEE, and Tessa Tielert, Member, IEEE

Abstract—The characteristics of vehicular communication environments and their networking and application requirements have led to the development of unique networking protocols. They enable vehicle-to-vehicle and vehicle-to-infrastructure communication based on the IEEE 802.11 technology, ad hoc principles, and wireless multihop techniques using geographical positions. These protocols, which are commonly referred to as Geocast, greatly support the vehicular communication and applications but necessitate a tailored security solution that provides the required security level with reasonable processing and protocol overhead, as well as reasonably priced onboard and road-side unit equipment. In this paper, we present the design of a security solution for Geocast, which is based on cryptographic protection, plausibility checks using secure neighbor discovery and mobility-related checks, trustworthy neighborhood assessment, and rate limitation. We analyze the achieved security level of the proposed scheme and assess its overhead and performance. Furthermore, we develop a software-based prototype implementation of a secure vehicular communication system. We find that the proposed security measures could result in a network performance bottleneck in realistic vehicular scenarios. Finally, we analyze the tradeoff between security overhead and protocol performance and determine the minimal processing overhead needed for acceptable performance.

Index Terms—Geocast, performance, secure, vehicular communication.

I. INTRODUCTION

EHICULAR short-range wireless communication is commonly regarded as the basis of future intelligent transportation systems (ITSs). Utilizing basic technologies such as the wireless local area network (WLAN), the global positioning system (GPS), digital maps, and others, vehicular communication protocols enable vehicles to exchange information with other vehicles, road-side infrastructure, or Internet nodes. Applications at the top of the communication protocol stack provide various services: *Safety applications* minimize road ac-

Manuscript received March 7, 2009; revised June 27, 2009 and December 20, 2009; accepted February 23, 2010. Date of publication March 15, 2010; date of current version June 16, 2010. This work was supported in part by the EU FP7 project "PRE-DRIVE C2X" under Grant number 224019 and in part by the German Research Foundation within the Research Training Group GRK 1194 "Self-organizing Sensor-Actuator Networks." The review of this paper was coordinated by Dr. L. Chen.

- A. Festag is with NEC Laboratories Europe, Heidelberg, Germany (e-mail: andreas.festag@neclab.eu).
- P. Papadimitratos is with the Institute of Communication Systems, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: panos.papadimitratos@epfl.ch).
- T. Tielert is with Decentralized Systems and Network Services Research Group, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (e-mail: tessa.tielert@kit.edu).
- Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2010.2045014

cidents and improve road safety, *traffic efficiency applications* optimize the vehicle flow on roads, and *infotainment applications* provide commercial, leisure, and convenience services.

In an ideal world, vehicular communication can provide great benefits to all road users and achieve a major step toward safer, cleaner, and smarter roads [1]. When vehicular communication is deployed in the real world, communication devices can malfunction, and adversarial users can misuse communication services. These issues can render the system useless, turn the benefits into hazards, and make system investments obsolete. Therefore, data security is commonly recognized as a mandatory and integral part of the system design. Nevertheless, some of the security objectives, particularly nonrepudiation, also pose privacy concerns for drivers [2].

Vehicular communication has specific characteristics and requirements, including intermittent access to a communication infrastructure and the need for self-organization, high node mobility, scalability with a number of nodes ranging from sparse to dense scenarios, information dissemination in geographical regions, the need for reliable data transmission with short delay, and fairness in resource usage. These characteristics and requirements have led to the design and development of specific communication protocols for vehicular ad hoc networks (VANETs). Networking protocols for multihop ad hoc communication exploiting geographical positions—referred to as Geocast—have been paid particular attention in the research community and are considered in the European standardization process [3], [4].

The Geocast protocol requires a security solution that is tailored to its specific characteristics, while meeting the requirements of secure vehicular communication systems. There has recently been significant progress on that front, and our investigation and solution are compatible with state-of-theart security architectures for vehicular communications (see Section V). Our objectives for securing Geocast fall into three categories: function, performance, and cost. The functional objectives mainly pertain to integrity, authentication, nonrepudiation, as well as (revocable) anonymity; specific networking aspects need to be considered due to the use of positions and timestamps in the Geocast protocol. The performance objectives concern quantitative demands on the number of security operations and the security packet overhead, as well as the delay for cryptographic processing and end-to-end (E2E) packet transmission. The cost objectives are related to the investments needed for a communication unit; they are linked to the functional and performance objectives of the proposed security solution. A tradeoff among the three objectives exists and can be adjusted. For example, we can minimize costs at the

expense of performance or lower the security level. Likewise, we can improve the performance at higher costs.

This paper makes the following main contributions: 1) design of a security solution of vehicular communication for ad hoc and multihop routing based on geographical addressing (Geocast); 2) assessment of the achieved security level based on a vulnerability analysis; 3) estimation of the protocol overhead caused by the security solution for state-of-the-art networking protocols; and 4) analysis of the tradeoff between functional security level, protocol performance, and associated costs for the communication unit.

The remaining sections of this paper are structured as follows. Section II summarizes the state of the art in security solutions for vehicular communication. Section III describes the system and protocol architecture and introduces the geographical routing. Section IV outlines the attacker model and a vulnerability analysis. Section V details the security solution for the geographical routing. Section VI analyzes the achieved security level. Section VII assesses the overhead and performance of the secure protocol based on results from the analysis, measurement, and simulation. Section VIII provides a discussion of the overhead and performance results with respect to the costs and security level. Section IX concludes this paper.

II. RELATED WORK

Major worldwide efforts are dedicated to vehicular communication based on WLAN technology, most importantly, the Vehicle Safety Consortium (VSC), the Vehicle Infrastructure Integration initiative (VII), and the IntelliDrive in the U.S.; the Car-2-Car Communication Consortium (C2C-CC) [3] and R&D projects for cooperative systems (CVIS, SAFESPOT, COOPERS, NoW—Network on Wheels, and SimTD) in Europe; and the Advanced Safety Vehicle (ASV) project and the Internet ITS consortium in Japan. The results of these initiatives are being adopted by standardization bodies (IEEE 802.11p and 1609, ETSI TC ITS, ISO TC204 WG16, and others).

Within these activities, security is increasingly being considered but still at an early stage. Among the recent R&D projects, *NoW—Network on Wheels* [5] and *SeVeCom* [6] specifically address the aspects of communication security and its integration into the vehicular communication system. Various research papers have been published: [7] and [8] present challenges of vehicular communication security, [9] and [10] systematically analyze potential attacks, and [11]–[15] describe general assumptions, requirements, and design principles to secure vehicular communication but do not consider a specific routing protocol.

Routing in vehicular networks based on principles for ad hoc and multihop communication and utilizing geographical positions is steadily gaining momentum research, development, and standardization. Its basic concept has been well known for some years. Originally, it was developed for efficient and scalable unicast communication for general mobile ad hoc networks (MANETs) [16], [17]. Its applicability for VANETs was studied (e.g., in [18]), and numerous extensions and enhancements were proposed. Today, position-aided routing, which is commonly referred to as Geocast, is being utilized for unicast,

broadcast/geographically scoped flooding, and anycast communication. Security schemes designed for MANETs and geographical routing cannot easily be applied to Geocast because they were designed for the different requirements of MANETs [19]. Similarly, the existing standardized security approach for vehicular communication as part of the IEEE 1609 standard is restricted to single-hop communication, and therefore, it neither supports multihop nor addresses the specific aspects of Geocast [20], [21]. Another work proposes the application of hop-by-hop (HbH) and E2E signatures for vehicular communication but is limited to layer-2 operation [22].

A few researchers have addressed secure geographical routing [12], [23]-[25], but they do not take all of the vehicular requirements into account. Reference [23] proposes a security extension for an advanced location service in MANETs. Reference [24] addresses secure position-aided unicast communication in MANETs. Reference [25] introduces a queryresponse messaging protocol for unicast communication in which geographical routing is applied to forward data from the source S to the destination D and every message is acknowledged by source routing from D to S to detect malicious neighbors. Reference [12] focuses on techniques for position verification to detect adversarial nodes in networks with geographical routing. Compared with these existing approaches, our proposed security solution is specifically tailored to a broader set of vehicular scenarios covering geographical unicast, broadcast/geographically scoped flooding, and anycast. It represents a comprehensive approach, covering attack analysis, solution design, and evaluation. Special emphasis is put on deployability; we have implemented the proposed scheme in a software prototype [26] and conducted experiments with automotive hardware platforms for vehicular communication. It is worth noting that our work presented here integrates our previous conceptual work on security in VANETs [27] and other techniques such as secure neighbor discovery (SND) [28], [29]. However, communication and location privacy, which is typically based on changing and revocable pseudonyms (e.g., [30]), is regarded to be orthogonal to the presented security approach.

III. COMMUNICATION PRINCIPLES AND ROUTING

Use cases for vehicular communication differ in requirements and communication principles. Safety and traffic efficiency applications typically disseminate information about events or other vehicles in the local vicinity or a certain geographical region. Vehicles either periodically broadcast short messages at a high frequency (the so-called beacons or heartbeats) or generate messages when they detect an event and distribute it by multihop communication in a certain geographical area (event-driven messages). Beacons need to arrive at the neighbor nodes with a specific maximum delay and a minimum reliability. Event-driven messages need a rapid, reliable, yet efficient dissemination in a geographical region, potentially via multiple wireless hops. In contrast, infotainment applications typically establish sessions and exchange unicast data packets, potentially in great numbers, bidirectionally, and over multiple wireless hops.

For addressing and routing using geographical positions, the concept of Geocast was developed. Providing efficient and scalable routing in vehicular scenarios, it represents the state of the art for routing in VANETs [3]. Geocast assumes that every node knows its geographical position, e.g., by GPS, and maintains a location table with the geographical positions of other nodes as soft state. It supports the addressing of individual nodes, as well as geographical addressing, which can be regarded as communication with a group of nodes that are located in a geographical region.

The core protocol components of Geocast are beaconing, location service, and forwarding. With beaconing, nodes periodically broadcast short packets with their ID, current geographical position, speed, and heading. On reception of a beacon, a node stores the information in its location table. The location service resolves a node ID to its current position. When a node needs to know the position of another node that is not available in its location table, it issues a location query message with the sought node ID, sequence number, and hop limit. Neighboring nodes rebroadcast this message until it reaches the sought node (or the hop limit). If the query is not a duplicate, the sought node answers with a location reply message carrying its current position and timestamp. On reception of the location reply, the originating node updates its location table. Forwarding basically means relaying a packet toward the destination, and we distinguish the following forms.

- Geographical unicast (GeoUnicast) provides packet transport between two nodes via multiple wireless hops. When a node wishes to send a unicast packet, it first determines the destination position (by location table lookup or the location service) and forwards the data packet to the node toward the destination, which, in turn, reforwards the packet along the path until the packet reaches the destination.
- 2) Geographical broadcast (GeoBroadcast) distributes data packets by optimized flooding, where nodes rebroadcast the packets if they are located in the geographical region ("dissemination area") determined by the packet. Advanced broadcasting algorithms ensure avoidance of the so-called "broadcast storms" and minimize overhead (see below). Geographical anycast (GeoAnycast) is similar to GeoBroadcast but addresses a single (i.e., any) node in a geographical area.
- 3) Topologically scoped broadcast (TSB) provides rebroadcasting of a data packet from a source to all nodes in its *n*-hop neighborhood. Single-hop broadcast is a specific case of TSB, which is used to send periodic messages. Each data packet carries the source's and previous forwarder's positions at the expense of a moderate packet overhead and updates its location table accordingly. Consequently, Geocast defines packet headers with fields for node identifier, position, and timestamp for source, sender, and destination, and other fields. In the header, we distinguish between immutable and mutable fields: Immutable fields are not altered in the forward-

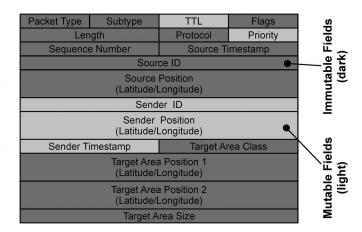


Fig. 1. Example Geocast packet with mutable and immutable fields.

ing process, while *mutable fields* can be updated by forwarders (see the *GeoBroadcast* header in Fig. 1 for illustration). An example of an immutable field is the destination area (target area positions 1 and 2 in Fig. 1), which is set by the source node and should, according to the protocol specification, not be changed by forwarders. In contrast, mutable fields of the packet header can be altered by a forwarder, such as the forwarder (i.e., sender) ID, position, and timestamp. Likewise, for other types of packet headers, forwarders can change the mutable fields on the fly, such as the destination position in a GeoUnicast if the forwarder has more recent information in its location table about the destination.

Another optimization is the use of feedback from the medium access control (MAC) protocol layer beneath the Geocast protocol to provide the link status to individual neighbor nodes. For example, a MAC "excessive retry" indicates that the connectivity to a node has been lost and that the corresponding entry can immediately be removed from the location table.

Specific forwarding algorithms ensure efficiency and scalability. For GeoUnicast, a node executes a greedy forwarding algorithm, which sends a packet to a neighbor with the minimum remaining distance to the destination. For GeoBroadcast, emergency message dissemination for vehicular environments (EMDV) [31] represents an algorithm that decreases and controls the number of packet retransmissions for directional flooding of a packet inside a geographical area. Basically, EMDV combines two mechanisms, namely greedy forwarding and contention-based forwarding. A node selects a next hop in the forwarding range² and forwards a data packet using the greedy forwarding toward a specified direction. Simultaneously to the packet processing in the next hop, all other nodes in the communication range also receive and cache the data packet, whereupon they start a retransmit timer that is indirectly proportional to the geographical progress. During the waiting time, nodes listen to ongoing transmissions and count the redundant retransmissions of the cached data packet. If the retransmit timer expires and the number of redundant retransmissions has not exceeded a predefined threshold, the

¹The originator of a message is referred to as the source, and the last forwarder is referred to as the sender.

²This refers to the area of the single-hop communication range with high packet reception probability.

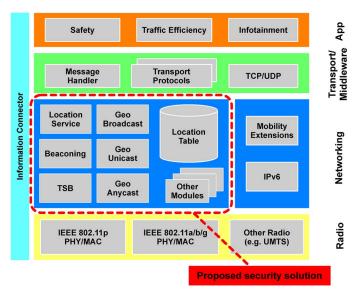


Fig. 2. Protocol architecture for vehicular communication.

packet is rebroadcast. If a node overhears transmissions and the retransmission counter exceeds the threshold, the node stops contention and does not forward the packet again. As a result, EMDV combines a rapid and reliable dissemination, where redundant retransmissions are controlled to reduce data congestion.

The Geocast protocol with its enhancements for efficiency and reliability is the networking protocol for which we propose our security solution (see Fig. 2).

IV. VULNERABILITY ANALYSIS

Network nodes either are correct (or honest) and comply with the protocol definition or are adversarial, i.e., they deviate from the protocol definition. We are primarily interested in active adversaries, because passive adversaries, i.e., eavesdroppers, can intercept packets and extract information about other network nodes³ but cannot affect or change their behavior.

Without the appropriate credentials and cryptographic keys, an adversary is *external*, i.e., it cannot create packets like those of a legitimate participant in the protocol execution. However, an external adversary can *replay* valid packets. The reception of replayed Geocast packets (data or control) can affect the protocol's operation: e.g., a replayed packet could mislead an honest node into thinking that it has a neighbor that is not actually present. External adversaries can also *jam* network traffic within their range and thus selectively or completely prevent communication; such attacks are relevant for any wireless communication system but are beyond the scope of this paper.

Nonetheless, the possession of credentials and cryptographic material does not necessarily guarantee a flawless protocol operation: Nodes can be compromised, infected, or deliberately misconfigured. Those *internal* adversaries can, in contrast to external adversaries, fully participate in the protocol execution.

Internal adversaries can create and inject any packet or modify packet fields to any arbitrary value, as long as (parts of) packets are not cryptographically protected by a correct node. Of course, internal adversaries can avoid any protocol action (e.g., packet transmission).

Multiple adversarial nodes can be present in the network. They can misbehave either independently or collusively, i.e., coordinate their actions by exchanging information. If adversaries share their private cryptographic keys, each of them can act as multiple nodes. However, in all cases, adversaries are computationally limited, that is, they cannot break correctly implemented cryptographic primitives. Finally, the adversary may only be capable of altering the inputs into the protocol with no control over the protocol functionality, i.e., it may find it easier to act as an input-controlling adversary, e.g., by compromising an onboard sensor.

An extensive discussion of adversary models for vehicular communication systems including the aforementioned models is available in [32]. Next, we discuss Geocast-specific attacks. In this case, the objective of the adversary is to degrade or prevent Geocast communication, i.e., the delivery of data, and we consider how the adversary can abuse the Geocast functionality to achieve this.

Destination location manipulation: The adversary can try to control the destination location information. This can be achieved by subverting the location service, if the service is used, or by impersonating the sought node and replying with false coordinates. Alternatively, an adversary that relays a location reply packet could modify it. In a more sophisticated attack scenario, the adversary could prevent the nodes that it targets from communicating with the location service or the querying nodes and then replay outdated replies from the actual node. It is worth noting that the adversary could have queried the node earlier.

Manipulation of their own position: The adversarial nodes always control the content of the packets they send. As a result, they can falsely advertise their position. This would affect the system's perception of their location and, consequently, their involvement in data forwarding. Similarly, when queried as a destination, an adversarial node can also manipulate its own position. Nonetheless, this would be of limited use if the adversarial node needed to receive data. If not, the adversarial node can lie about its position to force neighbor and nonneighbor nodes to change their location tables. In a different twist, an adversarial node can only locally advertise false locations so that only its neighbors falsely update its location. If the adversary attempted to impersonate other nodes, then a type of Sybil attack could locally be mounted (with each fictitious node at a distinct purported location).

Manipulation of other nodes' positions: Beyond interfering with the location service, the attacker could forge or modify the location information reported in other in-transit packets. That is, the adversary could manipulate the position of other nodes by modifying their coordinates in data packets to cause false updates of the location tables of subsequent forwarders. The adversary could also forge

³Thwarting those attackers is important in terms of privacy (e.g., location privacy), but this problem is orthogonal to our work.

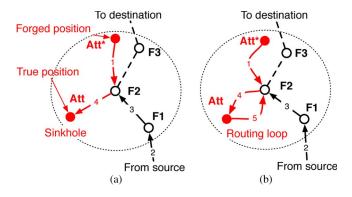


Fig. 3. Attack on geographical routing. (a) Sinkhole attack. (b) Routing loop attack.

other nodes' beacons: If the victim node is present in the same neighborhood, the impact of the attack would be low, as the legitimate control traffic would "correct" the attack. However, the adversary can relay beacons of other nodes and "inflate" the neighborhood of legitimate nodes (i.e., mislead them into thinking that they are neighbors with nodes that are, in fact, remote).

Data-plane attacks: The modification of data packet payload or packet dropping can also disrupt communication. To do so, an adversarial node must become part of the path along which the packet is forwarded. For that purpose, it attacks at the control plane to falsely present itself as being the most favorably placed, among their neighbors, to relay data packets. Fig. 3(a) illustrates such nodes discarding packets they receive to forward. From a different point of view, control-plane attacks can result in loops or an inability to bridge "gaps" on the network graph via the use of greedy forwarding recovery mechanisms. A situation with a loop created by misbehavior is shown in Fig. 3(b). In yet another scenario, the adversarial nodes could seemingly abide by the protocol yet forward data packets to suboptimal (i.e., not the closest to the destination) next-hop nodes. This would cause packets to travel longer in the network and perhaps even become lost when their time-to-live value is exhausted.

Clogging denial-of-service attacks: There is one such attack that relates to Geocast. It allows an adversary to transmit a single packet but causes many spurious packets to be sent across the network. To achieve this, the adversary can utilize the communication mode that allows data distribution within a geographical area. Moreover, a high-rate injection of control or data packets can consume bandwidth and processing power and disrupt the system if the attacker can abuse the processing resources of correct nodes.

V. SECURE GEOCAST

For our security system, we design mechanisms to safeguard the functionality of Geocast, relying on *cryptographic primitives*, *SND*, and *mobility-related plausibility checks*. The latter two mechanisms contribute to a *trustworthy neighborhood assessment* that rates nodes as truthful (or not) neighbors. In addition, we introduce a mechanism for the *rate limitation* of

data traffic. Since Geocast allows an adversary to incur overhead on N nodes by sending a single packet, the rate limitation can mitigate clogging denial-of-service attacks. Fig. 4 gives an overview of the security operations that will be explained later in this section. The figure also explains the sequence of security operations for a forwarding node. Compared with a forwarding node, a source node applies only the last three steps (steps 7–9). Additionally, it generates and attaches a signature over the immutable field and appends the source certificate. A destination node executes all but the last three steps (steps 1–6). If any of the checks fails, the packet is discarded, and the subsequent steps are not executed.

We assume the presence of a certification authority (CA), which manages credentials for all entities in the vehicular communication system and enables the use of public key cryptography. The CA interacts with the VANET through the road-side communication infrastructure. For details of the CA operation and how it communicates with the vehicular communication system, we rely on common and prominent architectures for use of public key cryptography in vehicular communication, notably the IEEE 1609, ETSI TC ITS, and C2C-CC, and R&D efforts such as SeVeCom and NoW-Network on Wheels [3], [11], [14], [20]. For Geocast, we assume that each node is equipped with a public-private key pair and a certificate, which binds its public key to attributes such as its role as a regular (private) vehicle, a public-safety vehicle, or a road-side unit (RSU). Without loss of generality, the key pair and certificate can be short lived, which is broadly accepted in the aforementioned approaches, so that pseudonymous authentication can protect the location privacy of regular vehicles.

A. Cryptographic Protection

To ensure protection, we utilize public key cryptography and digital signatures to enable secure communication without prior association (e.g., key establishment).

Periodic Packet: For *beacons*, a single signature is calculated, and a certificate is attached. The source node signature covers the entire Geocast packet. This is straightforward since intermediate nodes are not expected to change header fields.

Event-Driven Packet: We propose a combination of HbH (neighbor-to-neighbor) and E2E (source-to-destination) security. Hence, we protect a multihop packet with two signatures: the *source signature*, which is calculated by the source over the immutable fields, and the *sender signature*, which is generated by each forwarding node over the mutable fields (see Fig. 5).

Upon reception of a packet, a forwarding node 1) verifies both the source and forwarder signature; 2) updates the mutable field values and generates a new sender signature; 3) replaces the old sender signature with the new one; and 4) reforwards the packet. The destination node only verifies the signatures from the sender and the source.

This approach is used for packets that propagate across multiple hops, i.e., *GeoUnicast* and *GeoBroadcast* (Fig. 2), as well as *location service query* and *reply* packets. For EMDV (see Section III), the cryptographic protection implies that signatures of overheard packets are also verified and that the redundantly retransmitted packet is signed.

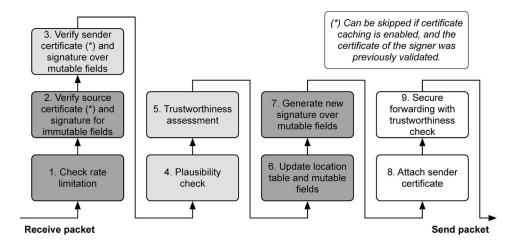


Fig. 4. Sequence of security operations for a multihop event-driven packet in a forwarder node.

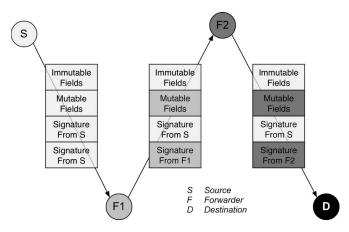


Fig. 5. Hybrid signature scheme.

Certificate Caching: To reduce the amount of cryptographic processing, a node can cache the certificate(s) carried in a received Geocast packet for the source and the sender (forwarding node). This mechanism requires that the certificate check (verification of the CA signature) is executed for an initial Geocast packet and is correctly verified before being saved. For subsequently received data packets, the node compares the received and the cached certificate. If they match, the newly received certificate does not need to be verified. The certificates are cached for a limited time, which is bound to the lifetime of node entries in the location table.

B. SND

To prevent the discovery of neighbors that are not actually neighbors, we integrate SND with the vehicular communication functionality, aiming at the least modification of the latter. More specifically, we want *correct* nodes to be able to ascertain whether the sender of a packet is indeed a *communication neighbor* [28], i.e., whether a correct node is able to directly communicate with (i.e., receive from) another correct node. Moreover, we leverage the existing vehicular communication functionality, rather than adding additional message complexity.

To achieve SND, as detailed and proved to be secure in [33], upon reception of a packet, a correct node determines its distance from the sender (fictitious or not) based on the location

and the time information that the packet carries. This information is already available in safety beacons in all architectures for vehicular communication: they carry the location and time at the sender, relying on the fact that all nodes are location aware and synchronized, notably by means of GPS. First, the receiver calculates the distance to the sender based on its own coordinates and those in the received packet. Then, it calculates the time of flight for the packet, i.e., the propagation delay from the sender to the receiver, by subtracting the received timestamp from the current time; the product of the time of flight and the radio-frequency (RF) propagation velocity gives a second distance estimate. If both estimates are approximately equal and below a neighborhood range R (essentially, at most the nominal communication range of the radio technology), the sender is declared a neighbor. The two most important practical aspects are 1) the availability of accurate time and location information and 2) the accurate setting of the timestamps and, to a lesser extent, locations. We explain how these issues are addressed next.

With the help of GPS, a fine time granularity and relatively precise location information is available. Currently, small-footprint and low-cost GPS receivers are commercially available, which achieve a low synchronization error e_t and low localization error e_{loc} . For example, GPS receivers such as those in [34] and [35] provide a timing accuracy within 10-15 ns and a localization accuracy of 6-30 m. With the RF propagation at 3.3 ns/m, the timing error can lead to distance estimate errors of up to 5 m. The SND protocol considers the cumulative effect of the two errors, with the two distance estimates within the sum of the two inaccuracies [29]; for the aforementioned example, the distance estimates within 5 plus 10 m. This leaves only a small amount space for misbehavior, in the sense that inaccuracies reduce the level of security, e.g., a node at a distance of R + 5 m could be accepted as a neighbor even though it is not. Overall, with a nanosecond time and meter position granularity, a secure neighborhood can be established with a high granularity, which is much less than the nominal vehicular communication ranges (from 200 to 1000 m with IEEE 802.11p).

It is also important that the timestamps are appropriately set for the packets used for SND. The timestamp value should be appended to the packet as close as possible to the time of transmission of the packet, i.e., the time at which the packet is transmitted by the MAC, avoiding any upper layer or other processing delays. The same is true for the location: the information in the packet should also be set at the time of transmission. However, the location accuracy is coarser grained than the time accuracy. The mobility of vehicles results in small displacements within small fractions of a second. For example, even a vehicle moving at 200 km/h would change its position by 5.5 cm within 1 ms. Thus, the challenge is to accurately set the timestamp.

As it is mandatory to digitally sign the beacon, the processing time for cryptographic operations must be taken into account. Moreover, the call to the cryptographic primitive is done at the MAC layer to avoid other delays in the protocol stack. Once the beacon is at the head of the MAC-layer transmit queue or, more specifically, when the beacon is about to be passed to the physical (PHY)-layer service access point, the timestamp field T is set to $T=t+T_{\rm crypto}+T_{\rm xmit}$, where $T_{\rm crypto}$ is the deterministic delay to calculate the digital signature and to append the certificate, and $T_{\rm xmit}$ is the remaining deterministic delay until packet transmission.⁴

To keep the implementation simple, we apply SND to beacons, more precisely to a protocol-selectable fraction of those beacons (or all beacons, for increased security). For those that we term *SND beacons*, the timestamp and location are set as explained earlier. This implies that the Geocast protocol passes the SND beacon to the MAC layer with its payload only, i.e., with the cryptographic fields left empty and the timestamp field not set. This way, the receiving nodes can obtain neighborhood verification for the senders of all SND beacons with no additional processing overhead compared with basic secure beaconing, at the expense of a slightly enhanced implementation.

Only the received packets that originate from a neighbor contribute to the maintenance of a node in the location tables. Note that location tables are also updated based on location information in packets other than beacons, i.e., data packets that contain the location of the forwarder. However, this is done only if the information in the Geocast header is newer than the information that was last used to update the neighbor's location and only if the neighbor was securely discovered will the nonbeacon packet be signed by the said node and the plausibility checks be successfully passed (which is discussed next).

C. Mobility-Related Plausibility Checks

Among nodes deemed as neighbors, mobility-related plausibility checks try to detect nodes that alter their own position to appear in favorable positions and attract traffic as forwarders of data. By evaluating successive beacons and data packets (that the adversary may forward), the checks attempt to find out whether a node frequently changes its claimed location to an extent that does not correspond to an actual vehicle. The checks utilize the position samples of neighboring nodes and, from these, calculate the velocity and acceleration of those nodes. The mobility parameters are verified to be in acceptable specific ranges determined by the situation, environment, and vehicle specifications, and they are compared with values from the receiver's local sensors. They can also be compared with the mobility parameters of the entire set of neighbors if sufficient redundancy is available. These checks go beyond the position verification approaches introduced in [12] but fit into their general framework of *autonomous* and *cooperative sensors*; what we present here in detail falls into the category of autonomous sensors.

The principle of mobility-related plausibility checks is based on calculations executed by the receiver to determine derivatives of the displacement for each neighbor (that is, every node that was previously discovered by SND). By calculating the velocity and acceleration, the receiver can detect spikes or discontinuities in displacement over a few successive samples. Even if an adversarial node could manage to maintain its velocity within acceptable limits, on the average (e.g., if it were inspected by neighboring nodes every minute or several seconds), frequent inspection over samples taken in close succession would reveal the inconsistent behavior. It is noteworthy that such frequent samples are readily available from the beacons (or heartbeats) transmitted by each vehicle; with a nominal value of a beaconing period of 100 ms (10 beacons/s), a position sample would be available for each neighbor at every fraction of a second (even under heavily congested scenarios, which could entail significant beacon loss).

In practice, the receiver calculates the distance between successive samples of the sender's position P_s at times t_n and t_{n-1} . The sender's velocity, i.e., the derivative of the displacement, is determined by the difference quotient of the displacement over the time, i.e., $v_s = \Delta P_s / \Delta t = d(P_{s,t_n}, P_{s,t_{n-1}}) / (t_n - t_{n-1})$. The receiver 1) checks whether the sender's velocity v_s is within the two extreme allowable values v_{\min} and v_{\max} ; 2) compares the sender's velocity with its velocity transmitted in the sender's beacon; and 3) compares v_s to its own velocity, which was obtained by onboard sensors (speedometer and GPS). The last check implies that if the difference is beyond a certain threshold, then it is impossible for the two nodes to continue being neighbors even though the falsified position data from the sender suggests that their physical distance has not diverged. If any of these three checks fail, the sender is deemed implausible. In addition to the velocity, the receiver uses a higher derivative of the displacement, i.e., the sender's acceleration $a_s = \Delta v_s/\Delta t = (v_{s,t_n} - v_{s,t_{n-1}})/(t_n - t_{n-1})$, and checks the value for the range and compares it with its own acceleration.

We illustrate the effectiveness of the mobility-related plausibility checks with the following scenario. An attacker M is located within the communication range of some victim V, and it may wish to simultaneously appear as the farthest node of V to be chosen as a forwarder in both directions of a road (back and forth). Assuming that V transmits a sequence of packets, e.g., every 500 ms, and the communication range is R=200 m, then M would need to swap the false positions

 $^{^4}$ It is possible to reduce T to microseconds: The T_{crypto} magnitude can be brought down from milliseconds to microseconds if a session (symmetric) key is used to authenticate the outgoing packet; the signature and public key encryption of the session key are done outside the critical path, i.e., before the SND call at the MAC layer. Then, the T_{xmit} is also on the order of a few hundreds of microseconds.

with a 400-m distance every 500 ms. The check in V will reveal an implausible velocity of 400 m/500 ms = 800 m/s = 2880 km/h.

D. Trustworthy Neighborhood Assessment

The *trustworthiness* expresses whether a node can be relied on as a truthful neighbor. Trustworthiness is assessed by plausibility checks, which impact the trustworthiness value of a node: plausible data in a packet increase trustworthiness, while implausible data decrease it. In practice, trustworthiness is a variable assigned to a node identifier and maintained in the node's location table. The trustworthiness value is defined between a minimum and a maximum value, i.e., C_{\min} and C_{\max} .

Starting from a predefined default value, trustworthiness is incremented/decremented for every executed plausibility check up to the maximum/minimum. The trustworthiness threshold determines the minimum value at which a node is considered to be trustworthy. When a node has a trustworthiness value smaller than this threshold, it is deemed *untrusted*. The following restrictions apply to untrusted nodes: 1) Packets originated or forwarded by an untrusted node are discarded, and 2) GeoUnicast packets destined for an untrusted node (as a next hop) are discarded.

Ignoring packets from untrusted nodes trivially implies that any information about other nodes carried in their Geocast header is also ignored. Overall, untrusted nodes are isolated by correct nodes. If relatively long-lived credentials are to be used, which is an aspect that is primarily related to policy decisions, then a reinstatement of the trustworthy neighborhood assessment can be envisioned. The reinstatement allows catering to benign faults. For example, if an onboard sensor in a node fails or a clock drifts in a given vehicle, the causing node would rightfully be ignored by other receivers. However, the same repaired vehicle encountered during the following day's commute should not simply be ignored. Similarly, the emergency braking of a vehicle may violate the mobility-related plausibility checks but should not necessitate "blacklisting" of the node. On the other hand, if credentials are short lived, the reinstatement of the trustworthiness assessment could be less useful or even irrelevant, for example, in the case of benign faults and pseudonyms with a lifetime of an hour or less. Overall, it is not our overall goal here to recreate a revocation scheme, and we therefore do not further analyze the technical and policy-related aspects. However, the detection of Geocast misbehavior could be added to such revocation schemes.

E. Rate Limitation

The rate control mechanism prevents the waste of resources caused by vast injection of false GeoBroadcast and TSB packets. If the rate of data packets originating from a node exceeds a predefined threshold, its packets are not forwarded any further. The mechanism is realized by means of digital signatures and the unique identification (timestamps) of the source and sender. To assess the actual data rate and smooth out short genuine bursts of data, we make use of the exponential weighted moving average (EWMA) for a neighbor node's data rate. The EWMA,

in fact, is an average of a series of collected data rate values that considers both the actual and older values. The EWMA calculation applies weighting factors that exponentially decrease, giving much more importance to recent observations while still retaining older observations. For tighter yet effective control, we define 1) a configurable smoothing factor for the weight in the EWMA calculation to control how agile to react when a node starts injecting data, 2) distinct rate thresholds for different node types (road-side station and private and public vehicles), and 3) the link of the packet transmission (e.g., the target geographical area) to different thresholds. For example, regular vehicles are not allowed to initiate Geocast packets beyond a given area size and low rate, while public vehicles may do so for larger areas and at higher rates.

VI. SECURITY ANALYSIS

Our secure Geocast protocol is resilient against external adversaries and thwarts a range of internal adversaries. In this section, we primarily discuss how Geocast-specific attacks are prevented. Attacks that do not abuse the Geocast protocol in particular but are generally applicable are out of scope; such attacks are jamming and clogging denial-of-service attacks and Sybil attacks. For the former, the rate-limiting mechanism of GeoBroadcast and TSB packets can limit their scope, while the latter can be thwarted by hardware security, as discussed briefly below.

Digital signatures and certificates prevent impersonation of other nodes. Without the proper credentials and cryptographic keys, adversaries cannot generate or modify messages in the name of other nodes. Impersonation of other nodes is possible only if an adversarial node holds the keys and credentials of other nodes. This could be the case if the attacker compromises the private keys of other nodes (e.g., physically extracts them from their memory) or if adversarial nodes share their private keys. In such a case, each node would act as a Sybil node, i.e., act as multiple nodes. This kind of attack is thwarted by the use of a tamper-resistant unit that performs cryptographic operations (e.g., the HSM of SeVeCom [11]): The adversarial nodes do not have access to correct nodes' private keys to share or cannot illegitimately extract keys of other nodes.

Digital signatures, which protect the integrity and provide authentication of Geocast control packets, can prevent the manipulation of destination-specific information: These signatures allow only the sought node to respond to a location query or update a location with a location service. Similarly, the manipulation of positions of other nodes is prevented: An adversarial node cannot modify fields in Geocast data or control headers that include location information for other nodes. As a result, internal or external adversaries are unable to manipulate the knowledge of correct nodes on the position of third nodes. Furthermore, the protection of data packets is clearly achieved with E2E and HbH cryptography.

A note on an alternative scheme that is apparently stronger than our hybrid scheme is given as follows. For an incremental signing scheme, the source would sign the whole packet, and each forwarder would copy the received mutable fields to the end of the packet, update its own mutable fields, and append a signature over the resultant entire packet. This scheme would allow the destination to authenticate every forwarder and thus maintain a larger network view and track changes of mutable fields. Nonetheless, this ability would not provide significant advantages in terms of security. The reason is that the Geocast operation is largely localized, i.e., the location of the neighbors and the destination are fundamental for the HbH forwarding process. Moreover, the additional signatures do not prevent an internal adversary from claiming any location as its own.

Message freshness is achieved by means of timestamps and geostamps (in general, the position of the sender, i.e., forwarding node, and not the source). At first, either replayed packets would have the sender's coordinates far beyond the perceived nominal range (distance) of the receiver or the coordinates would be correct but correspond to a sender that was at the given location at some point in the past; in this case, the received message timestamp would not be in sync with the receiver's clock. Such rigid validation is possible because nodes are location aware and synchronized.

Location awareness, particularly the availability of a node's own correct location, and synchronization are achieved by GPS, within the system errors (see Section V). The unavailability of GPS or attacks against GPS are beyond the scope of this paper. Future positioning systems will offer authentication services, e.g., as is the case for the upcoming Galileo system. Moreover, schemes like those described in [36] can detect spoofed replayed signals and prevent nodes from utilizing a time and location reference that is under the adversary's control.

With the location and time information, SND is possible: any two correct nodes can verify whether they are neighbors (if indeed they are) [29], [33]. This is critical for the availability of Geocast-based communication: An external adversary cannot relay beacons and mislead nodes into forwarding traffic to a next-hop node that is unreachable. Nonetheless, internal colluding adversaries could defeat the SND we employ here, as explained in [29] and [33]. In our context, internal adversaries can falsify the location and time values they enter in their beacons. Consider, for example, an adversarial node M with a location loc_M , a correct node U out of M's range, and another adversarial node N that is U's and M's neighbor. M could choose some fictitious location closer to U and a timestamp t' > t, with t being the correct time value, and sign the beacon; then, M could transmit with higher power than correct nodes and possibly get its message received at U or send it to N, which would relay it without any modification to U. As explained in Section V, additional complex functionality could thwart such attacks, yet our objective is the simplest in terms of functionality, which is still proven secure by SND [29], [33].

Internal adversaries verified as neighbors could declare false locations (and transmission times) for themselves, appearing this way as more attractive forwarders (i.e., closer to the destination). Consider an adversary M and a falsified position $loc(M,t_1)$ at time t_1 ; M may attract messages from a given node(s) that transmits traffic to a certain destination, as long as the misled correct node's position does not significantly change. However, if M wished to attract other flows, e.g., to a destination in the opposite direction, it would need to claim another "appropriate" fake position $loc(M,t_2)$. However, this

new choice of a fake location is mitigated by the plausibility checks: M can "adjust" loc(M) with respect to the positions of its neighbors only within the limits of what is deemed plausible. If $v_M = (loc(M,t_2) - loc(M,t_1))/(t_2-t_1)$ is such that it exceeds the maximum allowed velocity, all of M's neighbors will reduce its trustworthiness and then label it as a neighbor to ignore. Similarly, with an additional location sample by M, its neighbors can check if M's acceleration is below the maximum allowable and again reduce its rating.⁵

An adversary cannot misuse the plausibility checks against correct nodes: No reputation information is exchanged among nodes, and thus, nodes cannot falsely accuse other nodes. Instead, the plausibility checks can reveal suspected nodes that attempt to manipulate their own location. These nodes are eventually locally blacklisted by each node that individually perceives violations of the plausibility checks. As a result, these adversarial nodes will be avoided when packet forwarding decisions are made.

An adversary cannot create a routing loop, because it cannot mount a Sybil attack. Of course, a loop can be formed and include *only* adversarial nodes, but this would be equivalent to the case of having the first attacker in the loop dropping the packet. An adversary might, under special conditions, trigger the rerouting of packets enforcing a recovery mode for greedy forwarding (e.g., the perimeter mode in [16]), even if a "greedy" route exists. However, this would only cause a (mild) lengthening of the E2E route for a packet.

VII. COST AND PERFORMANCE ASSESSMENT

To investigate feasibility and deployability, we assessed the costs of the security solution in terms of the number of cryptographic operations, data, and processing overhead and studied the network performance of the secure Geocast compared with that of the basic scheme without security. For the cost and performance assessment, we apply a combination of analysis, measurement, and simulation.

The measurement setup comprises two hardware platforms. STANDARD is an IBM Thinkpad T41 notebook, with an Intel Pentium M 1.6-GHz CPU, 256-Mb (333-MHz) DDR RAM, and the Linux operating system. EMBEDDED is a NEC prototype for vehicular communication composed of a MIPS architecture VR5500 RISC CPU (400 MHZ), 64-MB DDR RAM, and the Linux operating system. Both platforms execute a security implementation as an enhancement of NEC's software prototype for vehicular communication [26], which is written in the C programming language, the OpenSSL cryptographic library, and the elliptic-curve digital signature algorithm (ECDSA) with a key length of 160 and 224 bits and Standards for Efficient Cryptography Group recommended 160- and 224-bit elliptic-curve-domain parameters for the signature algorithm for the node and the CA signature, respectively.

For the simulation-based assessment, we utilized the network simulator ns-2⁶. ns-2 is a packet-level discrete-event simulator

 $^{^5}$ Recall that there will be numerous location samples collected by default based on surrounding nodes' and, thus, M's transmissions (beacons and other control traffic).

⁶http://www.isi.edu/nsnam/ns/.

	Source	FORWARDER	DESTINATION	TOTAL
Secure Geocast (w/o CertCaching)	2G	4V + G	4V	2G + n(4V + G) + 4V
Secure Geocast (w/ CertCaching)	2G	2V + G	2V	2G + n(2V + G) + 2V
Hop-by-hop	G	2V + G	2V	G + n(2V + G) + 2V
End-to-end	G	2V	2V	G+2(n+1)V

 $\label{table I} \textbf{TABLE} \quad \textbf{I} \\ \textbf{Number of Cryptographic Operations for an Event-Driven Message} \\$

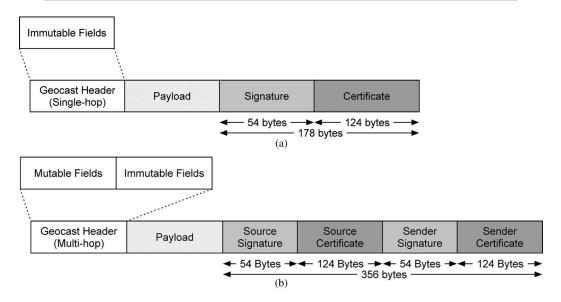


Fig. 6. Security trailer. (a) Periodic message. (b) Event-driven message.

that uses a centric event scheduler to handle events such as transmission, forwarding, and reception of packets, expiration of timers, and others. Beyond the event scheduler, ns-2 implements various network components and protocols. The core components of our simulation model are modules for the PHY, MAC, and networking layers that comprise individual nodes, as well as modules for the wireless channel, mobility patterns, and data generation. The simulations in this paper use version 31 of ns-2 with extensions for improved PHY- and MAC-layer modules [37], periodic messages, EMDV [31], and security (see Section VII-G1).

A. Number of Cryptographic Operations

We first analyzed the number of cryptographic operations required to process an event-driven message in a source, forwarder, and destination node and, in total, for a forwarding chain with n forwarders. Results are listed in Table I for secure Geocast with and without certificate caching. In the table, G refers to a signature generation operation, and V refers to a verification operation pertaining to either a certificate or signature. For comparison, the table also includes a HbH and E2E scheme.

B. Security Data Overhead

The data overhead results from the security trailer (see Fig. 6) carried by every data packet. The trailer is composed of a signature and a corresponding certificate. A single trailer with a certificate of the source node and a signature calculated over

TABLE II
PROCESSING TIME FOR CRYPTOGRAPHIC OPERATIONS ON A
PERIODIC MESSAGE (MEASUREMENTS)

	STANDARD Platform		EMBEDDED Platform	
Processing time [ms]	MEAN	STD.DEV.	Mean	STD.DEV.
SHA1 generation ECDSA sign	0.004 2.938	0.001 0.038	0.035 16.755	0.003 0.266
Generation total T_G	2.943	0.038	16.856	0.235
SHA1 generation Certificate verification Key extraction ECDSA verification	0.004 3.980 0.119 3.524	0.001 0.005 0.004 0.037	0.025 22.600 0.141 22.348	0.003 0.147 0.008 0.237
Verification total $\approx 2T_V$	7.663	0.040	45.381	0.379

the whole packet results in an overhead of 178 B [see Fig. 6(a)]. An event-driven message carries two security trailers with a total size of 356 B: One with a source certificate and signature calculated over the immutable fields and the other with a sender certificate and signature over the mutable fields [see Fig. 6(b)]. For a payload size of 500 B, the data overhead amounts to 34% for periodic and 71% for event-driven messages (excluding the protocol-specific Geocast header).

C. Processing Time for Cryptographic Operations

We instrumented our software prototype to measure the processing time of the individual cryptographic operations on both hardware platforms. Table II lists the results for the cryptographic operations on the sending side (*generation total*) and the receiving side (*verification total* includes both the certificate and the signature verification). The measured suboperations

TABLE III
PROCESSING TIME FOR CRYPTOGRAPHIC OPERATIONS ON AN
EVENT-DRIVEN MESSAGE (MEASUREMENTS)

Processing time [ms]	Mean	STANDARD Platform MEAN STD.DEV. MIN.VALUE MAX.VALUE			
Source	5.425	0.052	5.304	5.550	
Forwarder	15.886	0.056	15,421	16.107	
Destination	12.942	0.058	12,731	13.087	

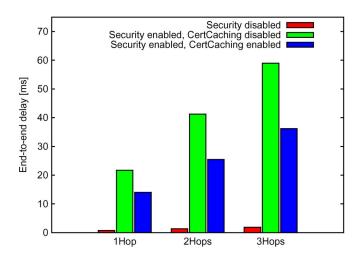


Fig. 7. E2E packet delay as a function of the number of hops (measurements).

show a dominant fraction for the signature generation, the signature verification, and the certificate verification, as detailed in the table. We also observed an increase in the processing time of about 500% for the EMBEDDED platform. For event-driven messages, the processing time is up to 15 ms for a forwarder (see Table III) measured in the STANDARD platform. For the EMBEDDED platform, we again assessed a rise of about 500%.

D. E2E Delay (Measurements)

In a next step, we created a simple laboratory setup with four nodes of the STANDARD platform, in which the nodes were positioned side by side, and enforced multihop forwarding of unicast packets, including the security processing from the source to the destination node via a forwarder. We indirectly measured the E2E packet delay by halving the round trip for communication over one to three hops. As expected, security considerably contributed to the E2E delay. The delay linearly increases to 60 ms for three hops, whereas certificate caching can reduce this delay to about 40 ms (see Fig. 7). The measured delay corresponds with the measured processing time for cryptographic operations in Section VII-C.

E. System Stability

The measurement results for cryptographic operations and E2E delay indicate that security processing has a considerable impact on the system performance. For further analysis, we model a node's security processing as an open queuing system of a single queue and a single server. We assume an infinite population and denote λ as the mean arrival rate and μ as the

mean service rate in the system. If we assume an infinite queue capacity, then the system is stable when λ satisfies

$$\lambda < \mu$$
. (1)

If the stability condition is not met, the number of packets in the system continuously grows, and the queue's response time and the packet delay become infinite.

The average packet processing time T_P can be calculated by

$$T_P = \sum_{x \in X} p_x \cdot T_x, \qquad \sum_{x \in X} p_x = 1 \tag{2}$$

where x represents the packet type from a set X, p_x is the probability that the packet type will occur, and T_x is the corresponding processing time. In the security queuing system, the set is made up of five packet types: generated and incoming beacons (BG and BI), generated and incoming event messages (EG and EI), and event messages to be forwarded (EF). Then, T_P can be expressed as

$$T_P = p_{BG}T_{BG} + p_{BI}T_{BI} + p_{EG}T_{EG} + p_{EF}T_{EF} + p_{EI}T_{EI}$$

 $P_{BG} + p_{BI} + p_{EG} + p_{EF} + p_{EI} = 1$ (3)

where $p_{BG}, p_{BI}, p_{EG}, p_{EF}$, and p_{EI} denote the probabilities of processing an incoming or generated beacon, incoming or generated event message, and an event message to be forwarded, respectively. $T_{BG}, T_{BI}, T_{EG}, T_{EF}$, and T_{EI} represent the corresponding processing times. They can be substituted by the processing time for cryptographic operations, i.e., either that for signature or certificate verification T_V (for simplicity, we use T_V as an approximation of the processing time for both operations) or that for signature generation T_G (see Table I), without certificate caching as follows:

$$T_P = (p_{BG} + 2p_{EG} + p_{EF})T_G + (2p_{BI} + 4p_{EI})T_V.$$
 (4)

If we take certificate caching into account and define p_H as the hit probability of the certificate cache, the term $4p_{EI}$ in (4) becomes $2p_Hp_{EI} + 4(1-p_H)p_{EI}$. Applying the same substitution to the terms $2p_{BI}$ in (4), we yield

$$T_P = (p_{BG} + p_{EF} + 2p_{EG})T_G + (2p_{BI} + 4p_{EI} - p_H(p_{BI} + 2p_{EI}))T_V.$$
 (5)

The lower and upper bounds of T_P are determined by the processing time for a generated beacon T_G ($p_{BG}=1$) and for an incoming event-driven message $4T_V$ ($p_{EI}=1$), respectively. Utilizing the measurement results from Table II, the minimum and maximum service rates $\mu=1/T_P$ correspond to 65 and 339 packets/s for the STANDARD platform and 11 and 59 packets/s for the EMBEDDED platform. If certificate caching is applied, the upper bound of T_P is reduced to $(4-2p_H)\,T_V$, which equals a minimum service rate of $\mu_{\rm min}=118~{\rm s}^{-1}$ (STANDARD) and 20 ${\rm s}^{-1}$ (EMBEDDED). For a packet arrival rate beyond this service rate, the system becomes unstable.

F. Acceleration Factor α

For further analysis, we introduce an acceleration factor α for the packet processing time of cryptographic operations T_P that is defined as the inverse scale factor for T_P

$$T_P^{\star} = \frac{1}{\alpha} \cdot T_P. \tag{6}$$

Taking the STANDARD platform as a reference, an acceleration factor of $\alpha=2$ would halve the measured values in Table II and allow for a maximum service rate of $\mu_{\rm max}=678$ packets/s; $\alpha=4$ even for 1.356 packets/s.

The acceleration factor takes into account the potential improvement of cryptographic operations due to the higher processing power of conventional CPUs or specialized hardware implementations. We particularly refer to advances in high-speed implementations of elliptic curve cryptosystems (ECCs) by means of high-performance scalar point multiplications in application-specified integrated circuits in recent years (see [38] for a survey of different implementation approaches) and to a recent report on ECCs implemented entirely in dedicated digital signal processor accelerator blocks of field-programmable gate arrays [39]. However, instead of considering state-of-the-art hardware platforms for cryptographic operations, we try to estimate how much processing capabilities a communication unit must have to guarantee a stable system. In this context, the acceleration factor α is used to assess the minimum acceleration needed compared with our prototype platform for vehicular communication with a software-based ECC implementation as the reference.

To estimate the order of magnitude of the node's required processing capability for a stable system, we assume a simple scenario with a high but typical density of stationary vehicles in which a vehicle is in the communication range of 100 other vehicles. For the analysis, we consider a sending rate of 10 beacons/s per vehicle, which results in a total arrival rate of incoming beacons of $\lambda=10^3~{\rm s}^{-1}$ at perfect reception probability. Compared with the maximum service rate (which is 339 and 59 packets/s for the STANDARD and EMBEDDED platforms), the beacon arrival rate considerably exceeds the processing capabilities of the platforms—not taking other packet types into account. Then, given the average packet arrival rate λ' in a particular scenario and the average service rate $\mu'=\alpha/T_p'$, the value of α , above which the system is stable, can be approximated as

$$\lambda' < \mu' \quad \Leftrightarrow \quad \alpha > T_P' \cdot \lambda'.$$
 (7)

In the scenario at hand, an acceleration factor $\alpha \approx 7.6$ would be needed to stabilize the system. Applying certificate caching and assuming a hit probability of the certificate cache $p_H=0.9$, α is still \approx 4.7.

We note that a finite queue capacity would also stabilize the system since the effective packet rate arriving at the vehicle would be limited by the queue length. However, packets are dropped if the queue size exceeds the capacity. As a consequence, the response time of the queue would be finite, but at the expense of higher packet loss and lower probability of reception.

G. E2E Delay (Simulations)

Simulation Setup: The results presented in the previous sections are based on simplifying assumptions about the number and distribution of nodes and their mobility, the wireless channel, and the protocols. All of these factors are expected to have a strong impact on the rate of packets to be processed. To study the security approach in more realistic environments, we conducted network simulations with ns-2. We chose EMDV (see Section III) for the dissemination of event-driven messages as a representative of the Geocast due to its promising performance [31].

In ns-2, we modeled the cryptographic operations as ns-2 nodes' timer-based packet queue, which queues incoming and outgoing packets until the node is able to process them. Then, the cryptographic operations to be performed for a packet causes the node to start a timer for the duration of the corresponding processing time that models the node's CPU being busy performing the cryptographic operation. Thus, the node stops processing packets for the duration of the timer. When all cryptographic operations for a packet have been (virtually) performed, the node resumes the packet processing. Furthermore, the ns-2 model also implements certificate caching as an optimization (see Section V) by extending the nodes' location tables to store certificates in addition to neighborhood information.

The studied scenario represents a typical German highway, using realistic car movement patterns derived from data provided by Daimler [40]. The chosen clipping is 8 km in length with three lanes per direction and 11 cars per lane and kilometer. All of the cars are assumed to be equipped with the communication system and comply with the system specifications, i.e., are trustworthy nodes. We configured the MAC and PHY layers according to IEEE 802.11p [37] and used the probabilistic Nakagami propagation model [41], [42], configuring the transmission power to correspond to a communication range of 1000 m under deterministic propagation. We set the beacon generation rate to 10 Hz. Considering a cooperative forward collision warning application, we chose one node to transmit EMDV event-driven messages once a second, which were to be disseminated within a distance of 2000 m from the source against its driving direction. For the EMDV-specific parameters (see Section III), we set the forwarding range to 500 m and the threshold for redundant retransmissions to 2. We assumed the payload for all packets to be 160 B, resulting in a packet size of 378 B for beacons and 556 B for event-driven messages when security is enabled. The packet size assumed a Geocast header of 40 B and 178- and 356-B security overhead for periodic and event-driven messages, respectively, as well as a 28-B MAC header. Furthermore, we parameterized the simulation with the processing time gained from measurements in Section VII-C for the STANDARD hardware platform as a reference. To study the effects of a possible hardware acceleration, we scaled the processing times in the simulation with the acceleration factor α (see Section VII-F). We simulated five different node setups with 20 seeds each. Table IV summarizes the simulation parameters.

Simulation Results: Comparing simulation results with and without security extension, we observed two major effects.

	TABLE	IV
CIMIL	ATION DA	DAMETEDS

Parameter	Value	
Scenario time and length	30 s, 8 km	
Lanes per direction, node density	3, 11 cars/km/lane	
Average node speed	121.86 km/h	
Propagation model	Nakagami (m = 3)	
Nominal communication range	1,000 m	
Data rate	3 Mbps (IEEE 802.11p)	
Beacon generation rate	10 Hz	
Beacon packet size	200 + 178 bytes	
Event-driven message packet size	200 + 356 bytes	
EMDV: Length of dissemination area	2,000 m	
EMDV: Forwarding range	500 m	
EMDV: Max redundant retransmissions	2	
Verification signature T_{SV} , certificate T_{CV}	3.52 ms, 4.10 ms	
Generation signature T_{SC}	2.94 ms	
Acceleration factor α	$1.5 \dots 5.0; 10^6$	

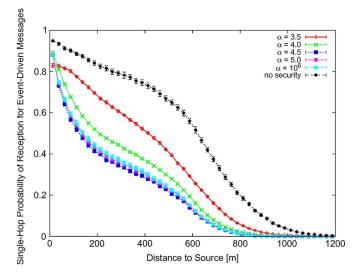
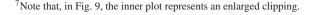


Fig. 8. Single-hop probability of reception over distance to the source for different values of the acceleration factor α (simulations, without certificate caching). The security packet overhead lowers the probability of reception as a result of channel saturation and more collisions in the medium access. $\alpha=10^6$ enforces a theoretical infinitesimal security processing time and shows the isolated impact of the increased packet size on the probability of reception.

First, the one-hop probability of reception decreases due to the larger packet size. In Fig. 8, this effect can be observed for $\alpha=10^6$, which enforces a theoretical infinitesimally small processing time. Since longer packets require more time to be transferred over the medium, the channel saturation increases, and collisions are more likely to occur, resulting in a lower probability of packet reception. However, the E2E probability of the reception of event-driven messages is not diminished and remains at 100% due to the robustness of EMDV.

Second, the one-hop transmission delay of beacons and the multihop E2E packet delay of event-driven messages significantly increase due to the security processing at each node. Fig. 9(a) depicts the average E2E packet delay for different acceleration factors α .⁷ It illustrates that, for $\alpha \leq 3.5$, the



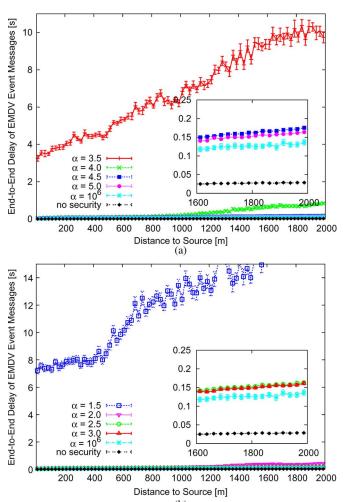


Fig. 9. E2E delay over the distance to the source for event-driven messages with different values of the acceleration factor α (simulations). (a) Without certificate caching. The simulation results show an acceptable E2E delay for $\alpha>4.0$. With slightly less acceleration of the security processing, the E2E delay rises by a magnitude. (b) With certificate caching. The E2E delay can be halved by certificate caching, but an acceleration factor of $\alpha>2.0$ is still required for an acceptable E2E delay.

average E2E delay at a distance of 2000 m is on the order of seconds. Between $\alpha=3.5$ and $\alpha=4.0$, a gap of one order of magnitude can be observed in the delay, whereas for $\alpha=4.0$, the average delay at a distance of 2000 m is 800 ms, and it is around 10 s for $\alpha=3.5$. For event-driven messages, the E2E delay accumulates over the wireless hops. In the given scenario, the dissemination area of 2000 m can be covered by forwarding packets over four to five hops on the average (see Fig. 10), whereas we count only the first packet that successfully arrives at the respective node. For an acceleration factor $\alpha>4.0$, the E2E delay almost linearly increases up to hundreds of milliseconds over seven hops at most. With smaller acceleration factors, the E2E delay is in the range of seconds.

The high E2E delay, even for a small number of hops, can be explained by the nodes' security queues. For an acceleration factor $\alpha \leq 4.0$, the average arrival rate of packets λ' exceeds the average processing rate μ' of the queue. As shown in Fig. 11(a), for $\alpha = 3.5$, the mean queue size—the number of packets in a node's security queue averaged over all nodes—rapidly grows

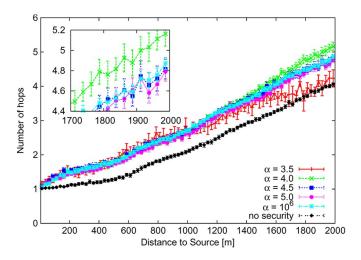


Fig. 10. Number of hops for event-driven messages over the distance to the source for different values of the acceleration factor α (simulations, without certificate caching). On average, four to five hops are needed to cover the dissemination area of 2000 m.

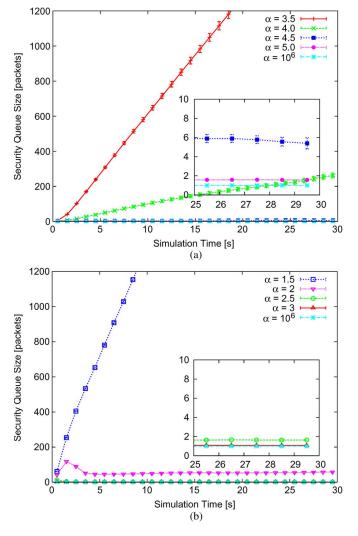


Fig. 11. Security queue size over simulation time with different values of the acceleration factor α (simulations). (a) Without certificate caching. For an acceleration factor $\alpha < 4.0$, the size of the security queue increases linearly, indicating an unstable system. (b) With certificate caching. An acceleration factor of $\alpha > 2.0$ is required to stabilize the system.

TABLE V AVERAGE SECURITY QUEUE COMPOSITION FOR $\alpha=3.5$ WITHOUT CERTIFICATE CACHING (SIMULATIONS)

Packet Type		Probability
Beacon Incoming Beacon Generation EMDV Incoming EMDV Forwarding EMDV Generation	p_{BI} p_{G} p_{EI} p_{EF}	$\begin{array}{c} 0.947285 \pm 0.000339 \\ 0.019349 \pm 0.000011 \\ 0.032573 \pm 0.000323 \\ 0.000786 \pm 0.000009 \\ 0.000006 \pm 0.000001 \end{array}$

over the simulation time, as do packet delays. For $\alpha=4.0$, the average number of packets in the queue still increases over simulation time, but for $\alpha=4.5$, the average size per node remains almost constant at about six enqueued packets.

Reconsidering the different packet types, we observed that, averaged over nodes, the relative frequencies of the occurrence of these packet types are close to constant. Table V presents the corresponding values for acceleration factor $\alpha=3.5$ with 95% confidence intervals. For different values of α , the average queue compositions only slightly vary. Furthermore, with enabled certificate caching and different values of α , the queue composition deviates by approximately 1%. In principle, this also shows that the vast majority of enqueued packets are incoming beacons, as the number of event-driven messages are negligibly small. Then, for the simulated scenario, T_P' in (4) can be approximated as

$$T_P' \approx 2p_{BI}T_V.$$
 (8)

Taking certificate caching into account, (8) can be rewritten as

$$T_P' \approx (2p_{BI} - p_H p_{BI})T_V \tag{9}$$

with p_h being the hit probability of the certificate cache p_H . Assuming $p_h = 0.92$ and $p_{BI} = 0.96$ as average values found by simulation, the total processing time T_P' is almost halved. This finding is reflected in the simulation results in Fig. 9(b). Comparing average E2E packet delays without and with certificate caching, at a distance of 2000 m, a delay of 1 s was observed for acceleration factor $\alpha = 4.0$ without certificate caching [see Fig. 9(a)]. A delay of about 0.5 s can be achieved for $\alpha = 2.0$ with certificate caching [see Fig. 9(b)]. Fig. 11(b) illustrates that the stability condition is met for $\alpha > 2.0$ as the average number of packets in the queue remains constant. In summary, the reduction in average processing time achieved by certificate caching has the same effect as applying hardware acceleration: μ is increased, stabilizing the system at a lower value of the acceleration factor α than that without certificate caching. Consequently, less acceleration would be required.

In the simulated scenario, λ' is approximately 566 packets/s. Thus, the stability condition in (7) yields that the system is unstable for acceleration factors of $\alpha \leq 4.16$ (disabled certificate caching) and $\alpha < 2.17$ (enabled certificate caching). These values correspond to our simulation results. Without certificate caching, $\alpha = 4.5$ yields feasible E2E delays, while $\alpha = 4.0$ does not. With certificate caching, the transition occurs between $\alpha = 2.0$ and $\alpha = 2.5$.

VIII. DISCUSSION OF PERFORMANCE RESULTS

We summarize the main findings of our cost and performance assessment as follows. Experiments with our software prototype for a secure vehicular communication system on standard computing hardware showed reasonable performance in terms of the protocol and processing overhead in simple scenarios. With embedded automotive-compliant hardware of only limited computing capability, we observed that the security overhead significantly increases the security processing time and, therefore, the E2E packet delay. We modeled the cryptographic processing as a queuing system. For a simplified setup, the numerical results indicate that the cryptographic processing results in system instability. To study the system for potential accelerated cryptographic processing and to give guidelines for the dimensioning of future onboard units, we introduced an acceleration factor α , which determines the acceleration for the service rate of the reference platform and a given packet arrival rate. In a next step, we conducted network simulations in realworld scenario settings with a realistic modeling of wireless channel characteristics, node mobility behavior, and advanced communication protocols. In particular, we studied the E2E packet delay and interpreted a growing delay over the simulation time as an indication of system instability. A proposed protocol optimization, i.e., certificate caching, almost halves a node's total security processing time but does not achieve the performance improvement required to stabilize the system.

Taking the proposed security solution design and its protection level as a premise, we can conclude three different deployment approaches to meet the performance, application, and system requirements: 1) use of high-performance general-purpose hardware for onboard units capable of fast security processing in pure software (this solution presumes that the cost–performance ratio of computing hardware exponentially improves—following Moore's law—and would result in powerful onboard units at reasonable costs in the near future); 2) deployment of a dedicated crypto-hardware that accelerates the execution time of cryptographic operations by large-size bit widths, parallelization of operations, and specific powerful arithmetic CPU cores; and 3) redesign of the application and networking protocols that sustainably reduces the number of packets, including generated and forwarded packets.

For the tradeoff between the security overhead and the protocol performance in approaches 1 and 2, our work provides a quantitative evaluation of the minimum required performance for cryptographic operations. For an acceleration factor $\alpha >$ 4.0, we observed a stable system and constant packet delays for multihop communication over a dissemination distance of 2000 m. For $\alpha < 4.0$, the packet-arrival rate exceeds the service rate of the system. We found that acceleration factors of $\alpha \geq$ 4.0 and $\alpha \geq 2.0$ without and with certificate caching, respectively, are needed. The third approach targets today's hardware, but as a consequence, system designers would need to abandon periodic high-frequency broadcast messages and Geocast over larger regions. Instead, mechanisms for strict access control, rate limitation, and data aggregation would have to be in place. We regard the three main approaches as a framework for the design of a security solution that will require a careful analysis of the technical and economic aspects in future work.

IX. CONCLUSIONS

We have proposed a security solution for vehicular communication networks that utilizes WLAN technology and ad hoc and multihop communication—more specifically Geocast—for safety, traffic efficiency, and infotainment applications. The solution is based on cryptographic protection, which involves a combination of HbH and E2E signatures and certificates. Complementary to the cryptographic protection, the solution provides plausibility checks, rate limitation, and malicious node isolation based on local trustworthy neighborhood assessment. Our analysis of the vulnerability and the achieved security level attests to the protection provided against the identified adversary models. A software prototype demonstrates the feasibility of the security approach. Our cost and performance assessment reveals high computing requirements for future onboard units and RSUs with a software-based security implementation. Our scalability studies using network simulations of realistic scenarios, environments, and protocols give quantitative measures for the dimensioning of future hardware units.

We have concluded that, for the deployment of the security solution with state-of-the-art communication protocols, a hardware acceleration at the expense of higher hardware costs is needed. Alternatively, a redesign of communication protocols to sustainably reduce the load in the network—and, implicitly, the security processing per node—can compensate the hardware needs.

ACKNOWLEDGMENT

The authors would like to thank C. Harsch, Dr. M. Killat, and Dr. F. Schmidt-Eisenlohr for their contributions, Prof. D. Westhoff, Dr. J. Härri, and Prof. H. Hartenstein for the discussions of the draft paper, and the reviewers for their valuable comments.

REFERENCES

- [1] Eur. Commission, Action Plan for the Deployment of Intelligent Transport Systems in Europe, Oct. 2008. [Online]. Available: http://ec.europa.eu/transport/its/road/action_plan_en.htm
- [2] J. P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security Privacy*, vol. 2, no. 3, pp. 49–55, May 2004.
- [3] Car-to-Car Communication Consortium, C2C-CC Manifesto, Sep. 2007, ver. 1.1. [Online]. Available: http://www.car-to-car.org
- [4] ETSI Technical Committee ITS. [Online]. Available: http://www.etsi. org/WebSite/Technologies/IntelligentTransportSystems.aspx
- [5] A. Festag, G. Noecker, M. Strassberger, A. Lübke, B. Bochow, M. Torrent-Moreno, S. Schnaufer, R. Eigner, C. Catrinescu, and J. Kunisch, "'NoW—Network on Wheels': Project objectives, technology and achievements," in *Proc. WIT*, Hamburg, Germany, Mar. 2008, pp. 123–128.
- [6] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for secure and private vehicular communications," in *Proc. ITST*, Sophia Antipolis, France, Jun. 2007.
- [7] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in Proc. HotNets-IV, College Park, MD, Nov. 2005.
- [8] J. Blum, A. Eskandarian, and L. Hoffman, "Challenges of intervehicle ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 347– 351, Dec. 2004.
- [9] J. Blum and A. Eskandarian, "The threat of intelligent collision," *IEEE IT Prof.*, vol. 6, no. 1, pp. 24–29, Jan/Feb. 2004.
- [10] A. Aijaz, B. Bochow, F. Dötzer, A. Festag, M. Gerlach, R. Kroh, and T. Leinmüller, "Attacks on inter vehicle communication systems—An analysis," in *Proc. WIT*, Hamburg, Germany, Mar. 2006, pp. 189–194.

- [11] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communications: Design and architecture," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 100–109, Nov. 2008.
- [12] T. Leinmüller, E. Schoch, and F. Kargl, "Position verification approaches for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 13, no. 5, pp. 16–21, Oct. 2006.
- [13] C. Tchepnda, H. Moustafa, H. Labiod, and G. Bourdon, "Securing vehicular communications: An architectural solution providing a trust infrastructure, authentication, access control and secure data transfer," in *Proc. AutoNet*, San Francisco, CA, Dec. 2006.
- [14] M. Gerlach, A. Festag, and T. Leinmüller, "Security architecture for vehicular communication," in *Proc. WIT*, Hamburg, Germany, Mar. 2007, pp. 119–124.
- [15] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng, "Cross-layer privacy enhancement and non-repudiation in vehicular communication," in *Proc.* WMAN, Bern, Switzerland, Mar. 2007.
- [16] B. N. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. MobiCom*, Boston, MA, Aug. 2000, pp. 243–254.
- [17] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *Proc. ACM Int. Workshop Discrete Algorithms Methods Mobile Comput. Commun.*, Seattle, WA, Aug. 1999, pp. 48–55.
- [18] A. Festag, H. Füßler, H. Hartenstein, A. Sarma, and R. Schmitz, "Fleet-Net: Bringing car-to-car communication into the real world," in *Proc. ITS World Congr.*, Nagoya, Japan, Nov. 2004.
- [19] E. Fonseca and A. Festag, "A Survey of Existing Approaches for Secure Ad Hoc Routing and Their Applicability to VANETS," NEC, Heidelberg, Germany, Tech. Rep. NLE-PR-2006-19, Mar. 2006.
- [20] Draft Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages, IEEE Std. P1609.2, 2006
- [21] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient VANET authentication," in *Proc. ESCAR*, Hamburg, Germany, Nov. 2008.
- [22] C. Tchepnda, H. Moustafa, H. Labiod, and G. Bourdon, "On analyzing the potential of a layer-2 multi-hop authentication and credential delivery scheme for vehicular communications," *Wireless Pers. Commun.*, vol. 51, no. 1, pp. 31–52, Sep. 2008. DOI: 10.1007/s11277-008-9607-x.
- [23] J.-H. Song, C. W. S. Wong, and V. C. M. Leung, "Secure position-based routing protocol for mobile ad hoc networks," Ad Hoc Netw.—Security Issues in Sensor and Ad Hoc Networks, vol. 5, no. 1, pp. 76–86, Jan. 2007.
- [24] S. Carter and A. Yasinsac, "Secure position aided ad hoc routing protocol," in *Proc. CCN*, Cambridge, MA, Nov. 2002, pp. 329–334.
- [25] V. Pathak, D. Yao, and L. Iftode, "Securing location aware services over VANET using geographical secure path routing," in *Proc. ICVES*, Columbus, OH, Sep. 2008, pp. 346–353.
- [26] A. Festag, R. Baldessari, W. Zhang, and L. Le, "CAR-2-X communication SDK—A software toolkit for rapid application development and experimentations," in *Proc. IEEE Veh. Netw. Appl. Workshop*, Dresden, Germany, Jun. 2009, pp. 1–5.
- [27] C. Harsch, A. Festag, and P. Papadimitratos, "Secure position-based routing for VANETs," in *Proc. VTC*, Baltimore, MD, Oct. 2007, pp. 26–30.
- [28] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun, and J.-P. Hubaux, "Secure neighborhood discovery: A fundamental element for mobile ad hoc networking," *IEEE Commun. Mag.*, vol. 46, no. 2, pp. 132–139, Feb. 2008.
- [29] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Secure neighbor discovery in wireless networks: Formal investigation of possibility," in *Proc. ACM ASIACCS*, Tokyo, Japan, Mar. 2008, pp. 189–200.
- [30] E. Fonseca, A. Festag, R. Baldessari, and R. Aguiar, "Support of anonymity in VANETs—Putting pseudonymity into practice," in *Proc.* WCNC, Hong Kong, Mar. 2007, pp. 3400–3405.
- [31] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-vehicle communication: Fair transmit power control for safety-critical information," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3684–3707, Sep. 2009. DOI: 10.1109/TVT.2009.2017545.
- [32] P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "Securing vehicular communications—Assumptions, requirements and principles," in *Proc.* ESCAR, Berlin, Germany, Nov. 2006.
- [33] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Towards provable secure neighbor discovery in wireless networks," in *Proc. ACM FSME*, Alexandria, VA, Oct. 2008, pp. 31–42.
- [34] Resolution-SMT GPS Receiver. [Online]. Available: www.trimble.com
- [35] LEA-5T Precision Timing GPS Module. [Online]. Available: http://www.u-blox.com/en/gps-modules/timing-modules/lea-5t.html

- [36] P. Papadimitratos and A. Jovanovic, "GNSS-based positioning: Attacks and countermeasures," in *Proc. MILCOM*, San Diego, CA, Nov. 2008, pp. 1–7.
- [37] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation architecture in NS-2," in *Proc. MSWIM*, Chania, Greece, Oct. 2007, pp. 159–168.
- [38] G. M. deDormale and J.-J. Quisquater, "High-speed hardware implementations of elliptic curve cryptography: A survey," *J. Syst. Archit.*, vol. 53, no. 2/3, pp. 72–84, Dec. 2007.
- [39] T. Güneysu and C. Paar, "Ultra high performance ECC over NIST primes on commercial FPGAs," in *Proc. CHES*, Washington, DC, Aug. 2008, pp. 62–78.
- [40] ĤWGui Project. [Online]. Available: http://www.informatik.uni-mannheim.de/pi4.data/content/projects/hwgui/
- [41] M. Nakagami, "The m-distribution, a general formula of intensity distribution of the rapid fading," in *Statistical Methods in Radio Wave Propagation*, W. C. Hoffman, Ed. Oxford, U.K.: Pergamon, 1960.
- [42] V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Poster abstract: Empirical determination of channel characteristics for DSRC vehicle-to-vehicle communication," in *Proc. VANET*, Philadelphia, PA, Oct. 2004, p. 88.



Andreas Festag (SM'08) received the Dr.-Ing. degree in electrical engineering from the Technical University Berlin, Berlin, Germany, in 2003.

He is currently a Senior Researcher with NEC Laboratories Europe, Heidelberg, Germany. His research is concerned with the architecture, design, and performance evaluation of wireless and mobile communication systems, currently focusing on vehicular communication.

Dr. Festag actively contributes to the CAR-2-CAR Communication Consortium and to ITS standard-

ization. He serves as chairman of European Telecommunication Standards Institute Technical Committee ITS WG3.



Panagiotis Papadimitratos (M'05) received the Ph.D. degree from Cornell University, Ithaca, NY, in 2005.

He was a Postdoctoral Fellow with the Virginia Polytechnic Institute and State University, Balcksburg. He is currently a Senior Researcher with the Institute of Communication Systems, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. He is the author of more than 70 technical publications on these topics and has delivered several tutorials, including ones at the Association

for Computing Machinery (ACM) International Conference on Mobile Computing and Networking and the ACM Conference on Computer and Communications Security (CCS). His research is concerned with security, networking protocols, and wireless and mobile systems.

Dr. Papadimitratos has served on the program committees of numerous venues, including the IEEE Conference on Computer Communications, the ACM International Symposium on Mobile Ad Hoc Networking and Computing, the ACM Conference on Wireless Network Security, the ACM Symposium on Information, Computer, and Communication Security (ASIACCS), and the ACM Workshop on Vehicular Ad Hoc Networks.



Tessa Tielert (M'10) received the Diploma degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 2008. She is currently working toward the Ph.D. degree in computer science within the Decentralized Systems and Network Services Research Group, Karlsruhe Institute of Technology.

Her research interests include mobile networks, simulation environments, and impact studies of vehicular communication.